

ОТЧЁТ ПО ВЫПОЛНЕННЫМ ЛАБОРАТОРНЫМ РАБОТАМ

QA-ENGINEER

Оглавление.

1. Выполнены работы и тесты по темам программы.
2. Перечень использованных инструментов и методов.
3. Методы проектирования тестов. Черный ящик.
4. Инструменты тестирования. Базы данных.
5. Тестирование веб-продуктов. Логи.
6. API.
7. DevTools.
8. Инструменты тестирования.
9. Приложение 1. Методы проектирования тестов на основе опыта.
10. Приложение 2. Изучение продукта и интеллект-карты.
11. Приложение 3. Попарное тестирование.
12. Дополнительный материал.

1. Выполнены работы и тесты по темам программы

(тесты в отчёт не вошли):

- Введение. Основы тестирования.
- Модели разработки ПО и место тестирования в них.
- Классификация видов тестирования.
- Архитектура ПО.
- Проектирование тестов. Тестирование требований.
- Изучение продукта и интеллект-карты.
- Методы проектирования тестов. Черный ящик.
- Методы проектирования тестов на основе опыта.
- Тестовая документация. Оформление тестов.
- Баг-трекер. Оформление дефектов.
- Инструменты тестирования. Базы данных.
- Тестирование веб-продуктов. Логи.
- API.
- DevTools.
- Инструменты тестирования. Боекомплект тестировщика.

2. Перечень использованных инструментов и методов.

Методы:

- Тестирование на основе чек-листов (checklist-based testing)
- Тестовый сценарий (test case)
- Тестирование на основе рисков
- Исследовательское тестирование
 - Чит-листы
 - Сессионное тестирование
 - Парное тестирование
 - Тест-туры
- Оракулы

- FEW HICCUPPS Майкла Болтона и Джеймса Баха
- Таксономия оракулов Дуга Хоффмана
- Принципы тестирования
- Задачи в тестировании
- Жизненный цикл разработки ПО
- Модели разработки ПО
- Классификация видов тестирования
- Тестирование требований и техники тестирования
- Исследование продукта
- Анализ тестирования
- Методы проектирования тестов
- Тестирование на основе чёрного ящика
- Классы эквивалентности
- Таблица альтернатив (решений)
- Диаграмма переходов и состояний (State Transition)
- Проектирование сценариев
- Интеллект карты
- UML диаграммы
- Agile (<https://agilemanifesto.org/iso/ru/manifesto.html>)
- Scrum
- Kanban
- Workflow задач
- Жизненный цикл дефектов
- Оформление баг-репорта
- Инструменты тестирования API
- REST-запросы
- Тест-дизайн для API
- Доступ и чтение логов

Инструменты:

- Средства виртуализации:
 - [VirtualBox](<https://www.virtualbox.org/>)
 - [VMware](<https://www.vmware.com/>)
- Скриншотеры, видеолоттеры:
 - [Monosnap](<https://monosnap.com/>)
 - [Sreenshoter](<https://skrinshoter.ru/>)
 - [Joxi](<https://joxi.ru/>)

- [Recordit](<https://recordit.co/>)
- Генераторы данных:
 - [Mockaroo](<https://www.mockaroo.com/>)
 - [Генератор случайных личностей](<https://randus.org/>)
 - [Временный email](<https://10minutemail.com/>)
 - [Генератор ИНН/ОГРН/ОКПО/СНИЛС](https://mobobe.com/wp-uploads/inn_ogrn/inn_ogrn.htm)
 - [Генератор GUID](<https://www.guidgenerator.com/>)
 - [Генератор случайных строк, цифр](<http://www.unit-conversion.info/texttools/random-string-generator/>) - и не только
 - [Perclip](<https://www.satisfice.com/blog/archives/downloads/tools>)
 - [Подсчет разных знаков в тексте](http://www.vipkontent.ru/?page_id=450)
 - [Генератор картинок](<https://lorempixel.com/>)
- Парное тестирование
 - [Pairwise testing](<https://www.software-testing.ru/library/testing/test-analysis/1304-pairing>)
- Файловые менеджеры:
 - [Far Manager](<https://farmanager.com/download.php?l=ru>) - после него никакой тотал командер не страшен
 - - [Total commander](<http://wincmd.ru/plugring/totalecmd.html>) - не боимся учиться им пользоваться, намного эффективнее проводника windows
- Сравнение текстов
 - - [Сравнение текстов онлайн](<http://text.num2word.ru/>)
 - - [Notepad++](<https://notepad-plus-plus.org/>) - классика, многофункциональный инструмент, много доп. плагинов, например - Compare, JSON Viewer, XML Tools, + Auto save.
- «Заполнители»
 - Bugmagnet(<https://bugmagnet.org/>) - плагин для Chrome, FireFox с предустановленными значениями полей разных типов из контекстного меню.
 - - FormFiller(<https://chrome.google.com/webstore/detail/fake-filler/bnjingeaknajbdcgpfkgnonkmiifhfo?hl=ru>) - плагин для Chrome, FireFox для автозаполнения больших форм
- Fiddler(<https://www.telerik.com/download/fiddler>)

- Pairwiser - это техника формирования наборов тестовых данных (<https://inductive.no/pairwiser/>)
- Docker
- DevTools - набор инструментов для отладки и тестирования UI.
- Postman - инструмент, который помогает формировать запросы и анализировать ответы со стороны сервера (касательно API).
- WinSCP
- MultyPutty
- MySQL
- JIRA
- Testlink
- xMind
- Draw.io
- Trello
- ChromeCacheView
- Swagger - инструмент API, который считывает и отображает информацию в спецификации (<https://swagger.io/>). Swagger. Petstore – образец приложения и инструмент демонстрации (<http://petstore.swagger.io>)
- Формат данных JSON
- Формат сериализации данных YAML

Теоретические знания:

- Стек протоколов TCP-IP
 - HTTP сообщения
 - Методы
 - Headers
 - Коды состояний
- Обзор протокола HTTP
- REST API
- Логирование

Технологии и инструменты, которые затронуты были косвенно:

- ELK stack. ElasticSearch — самонастраивающаяся база данных с хорошим поиском по тексту.
- ELK stack. LogStash — синхронизатор с ElasticSearch и сбор логов из источников.
- ELK stack. Kibana — веб-интерфейс с большим количеством дополнений.
- JavaScript – язык программирования.

- Grep – unix команда, которая позволяет по регулярному выражению найти все нужные строки в текстовом файле.
- Регулярные выражения (regex101.com, regexr.com, regexper.com)
- Скрипты (bash)
- Изолированное тестирование
- Проблемы тестирования микросервисов

3. Методы проектирования тестов. Черный ящик.

СЦЕНАРИИ:

Основной сценарий – все ок.

Проектирование сценария:

Название: Заполнение формы «Заказ расчета персонального финансового плана».

Предусловия: форма не заполнена, выбраны некоторые пункты по умолчанию.

Постусловия: данные переданы финансовому консультанту

Действующие лица: клиент, система

Триггеры: кнопка «Заказать расчет»

Сценарий:

Краткое описание:

Клиент заполняет форму

Иницируется отправка расчета

Результат:

Система принимает данные, отправляет отчет в CRM, CRM возвращает ответное сообщение и индикатором звонка, и результатом операции.

Альтернативный сценарий №1– «Сумма инвестирования» не заполнена.

Краткое описание:

Пользователь не заполнил форму с суммой инвестирования

Иницирует отставку данных из формы

Результат:

Система не принимает данные, отображает сообщение об ошибке «Сумма инвестирования должна быть больше, чем 0»

Альтернативный сценарий №2– указана «Дата реализации цели» меньше текущей + месяц.

Краткое описание:

Пользователь указывает дату меньше текущей + один месяц

Результат:

Отображается сообщение об ошибке «Дата реализации цели должна превышать сегодняшнюю не менее, чем на месяц».

Таблица состояний:



4. Инструменты тестирования. Базы данных.

Задание:

- 1) Напишите запрос, который выведет все обязательные параметры существующих типов персональных документов

```
SELECT * FROM gfo_personal_doc_prm_list WHERE required=1
```

- 2) Напишите запрос, который вернет количество уникальных клиентов из справочника персональных документов пользователей - GFO_PERSONAL_DOCUMENT

```
SELECT COUNT (DISTINCT clientexternalid) AS amount FROM GFO_PERSONAL_DOCUMENT;
```

- 3) Напишите запрос, который добавит новую запись в справочник типов персональных документов (GFO_PERSONAL_DOC_TYPE) – данные в запросе на свое усмотрение


```
INSERT INTO gfo_personal_doc_type (code, name, personal_doc_category_id, sequence)
VALUES (12, archive, 4, 1)
```

- 4) Напишите запрос, который удалит все изображения персональных документов, загруженные ранее 10 октября 2005 года

```
DELETE FROM gfo_personal_document_image WHERE upload_date < '20051010'
```

- 5) Напишите запрос, который отсортирует справочник изображений персональных документов по дате и времени изменения статуса записи в порядке убывания

```
SELECT * FROM gfo_personal_document_image ORDER BY status_change_date DESC
```

- 6) Напишите запрос, который обновит все незаполненные значения параметра по умолчанию на значение «TEST» в справочнике списка параметров существующих типов персональных документов - GFO_PERSONAL_DOC_PRM_LIST

```
UPDATE gfo_personal_doc_prm_list SET 'TEST' WHERE default_name IS NULL
```

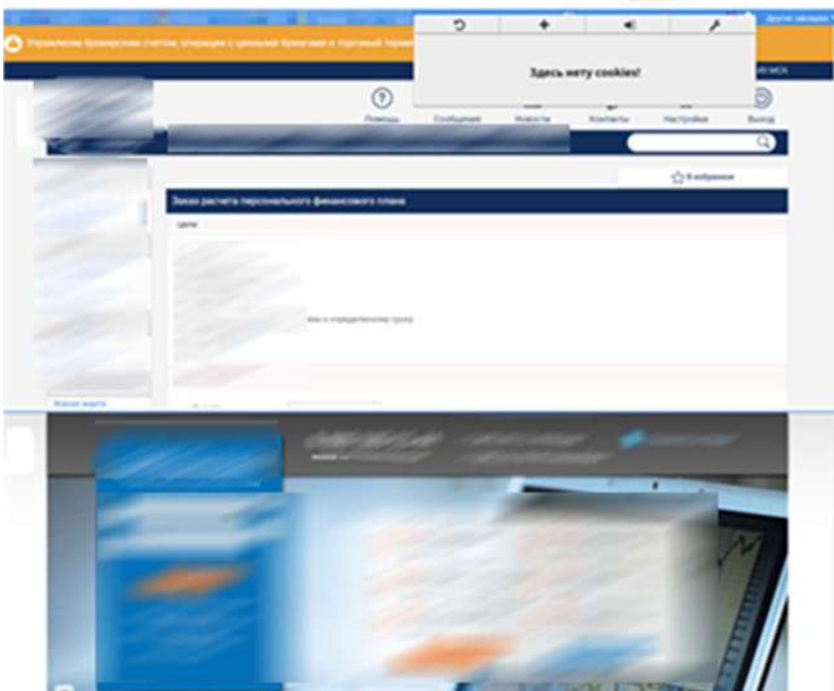
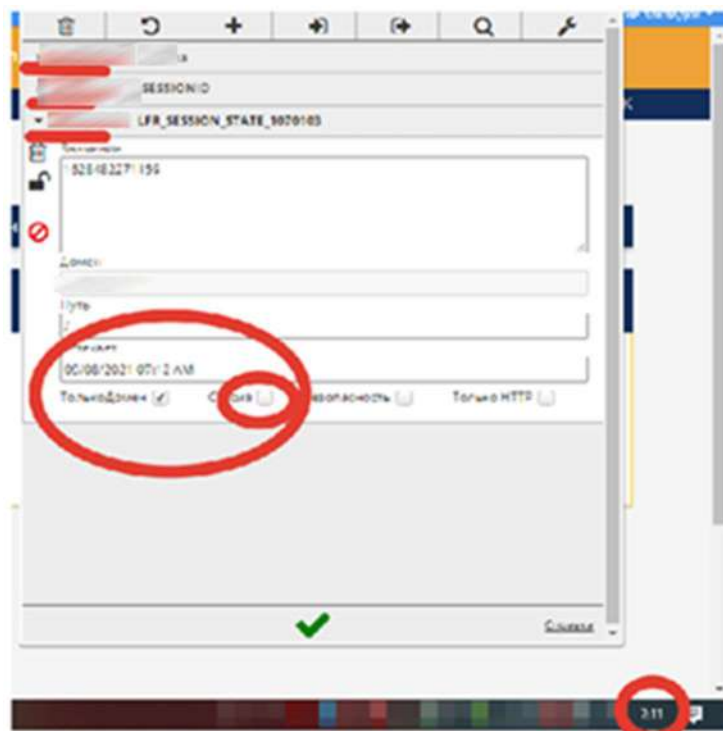
5. Тестирование веб-продуктов. Логи.

Задание 1. EditThisCookie

1. С помощью плагина **EditThisCookie** в Chrome изменить cookie пользовательской сессии на портале ABC, так, чтобы при выполнении любого действия на странице сессия пользователя считалась истекшей
2. Приложить скриншот изменения cookie, **описать полученный результат.**

Решение:

Авторизоваться на сайте, оставить время жизни cookie 1 минута, подождать 1 минуту, при переходе на другую страницу сайта (обновлении страницы) теперь требуется повторная авторизация, так же, как и при закрытии браузера, когда сеансовые cookies удаляются самостоятельно. Удаление вручную cookie сразу, а не выставление время жизни даёт тот же результат.

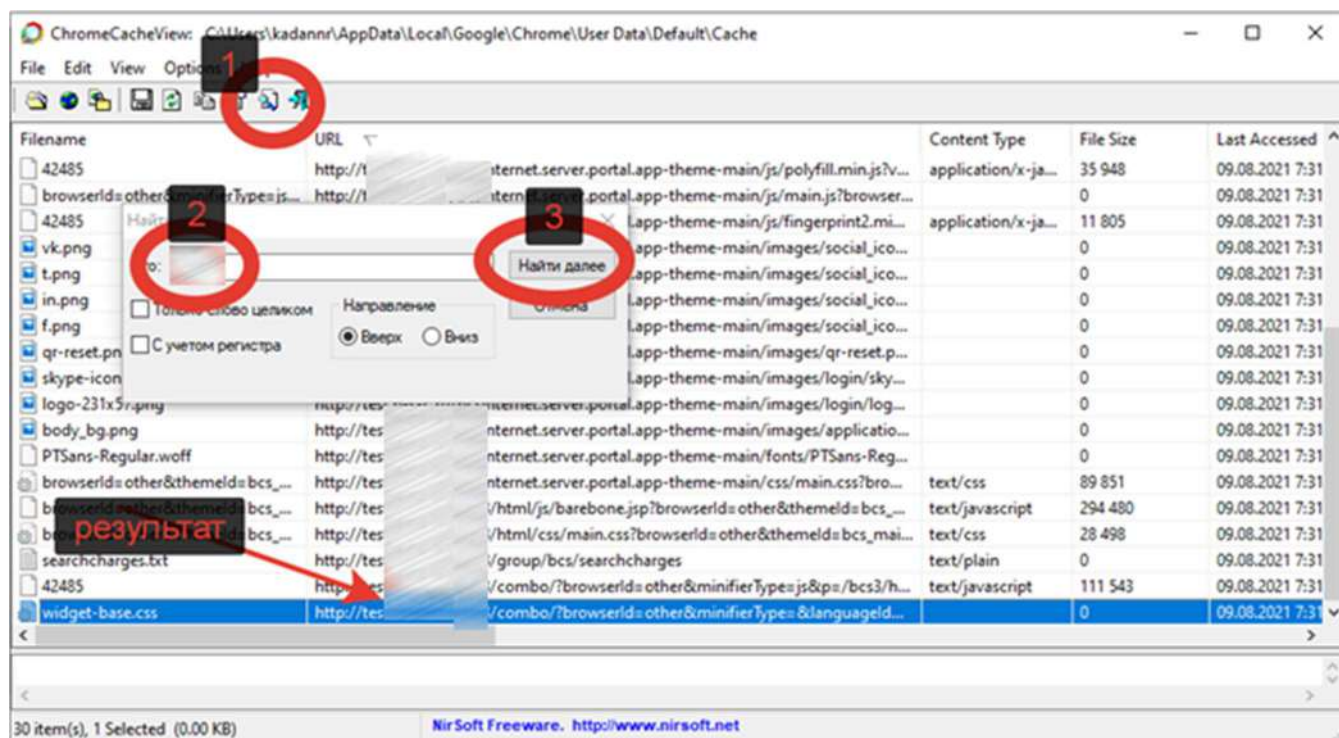


Задание 2. ChromeCacheView

1. С помощью приложения ChromeCacheView исследовать кэш, который сохранился при работе с порталом
2. Сделать поиск по названию сервера a-b-c и сделать скриншот поиска для отчета
3. В закешированных данных найти любую картинку с сервера wasrv и открыть ее в браузере - сделать скриншот из браузера

Решение:

Скриншот поиска сервера:



Задание 3. Логи

1. Изучить структуру логов портала.
2. Найти логи портала за текущее число (файл/-ы) - сделать скриншот
3. На Портале выполнить сценарий:
 1. Неуспешная авторизация (некорректный пароль) - приложить выдержку из лога по неуспешной авторизации
 2. Успешная авторизация - приложить лог с момента успешной авторизации до полной загрузки домашней страницы
 3. Отправить заявку на расчет персонального финансового плана - в логах найти запрос с своей заявкой (ориентир - вызов сервиса client_service_request), найти ответ, приложить скриншот заявки с портала и лог по выполненной операции на своем клиенте

Решение:

2. Скриншоты логов портала за текущее число:

[illegible]

Index of /logs/ara/portal

| Name | Last modified | Size | Description |
|------------|-------------------|------|-------------|
| 1-08-0... | 03-Aug-2021 22:58 | 9.6M | |
| -08-0... | 04-Aug-2021 23:00 | 3.4M | |
| 08-0... | 05-Aug-2021 18:30 | 11M | |
| 08-0... | 06-Aug-2021 20:39 | 13M | |
| -08-0... | 07-Aug-2021 22:43 | 3.9M | |
| -08-0... | 08-Aug-2021 23:33 | 7.0M | |
| 08-08-0... | 09-Aug-2021 08:03 | 432K | |

IBM HTTP Server

3 а. Выдержка логов при неуспешной авторизации:

08:02:09.862 [WebContainer : 2] INFO c.b.g.c.u.m.i.UsernamePasswordCredentialsManagerImpl
RID:c780e685-0e25-405d-a1c2-7a47d96c84d6 SID:- _DVD-WQY46pfTXZ5RxAX3s UN: - Channel =
IBS

08:02:09.863 [WebContainer : 2] INFO c.b.g.c.u.m.i.UsernamePasswordCredentialsManagerImpl
RID:c780e685-0e25-405d-a1c2-7a47d96c84d6 SID:- _DVD-WQY46pfTXZ5RxAX3s UN: - Login =
alifanovea

08:02:09.865 [WebContainer : 2] WARN c.b.g.c.u.m.i.UsernamePasswordCredentialsManagerImpl
RID:c780e685-0e25-405d-a1c2-7a47d96c84d6 SID:- _DVD-WQY46pfTXZ5RxAX3s UN: - Active
credentials for user 'alifanovea' not found.

08:02:09.866 [WebContainer : 2] WARN c.b.g.c.u.m.i.UsernamePasswordCredentialsManagerImpl
RID:c780e685-0e25-405d-a1c2-7a47d96c84d6 SID:- _DVD-WQY46pfTXZ5RxAX3s UN: - Blocked
credentials for user 'alifanovea' not found.

08:02:09.866 [WebContainer : 2] WARN c.b.g.c.u.m.i.UsernamePasswordCredentialsManagerImpl
RID:c780e685-0e25-405d-a1c2-7a47d96c84d6 SID:- _DVD-WQY46pfTXZ5RxAX3s UN: - Common
exception for username alifanovea null

08:02:09.866 [WebContainer : 2] WARN c.b.g.c.u.m.i.UsernamePasswordCredentialsManagerImpl
RID:c780e685-0e25-405d-a1c2-7a47d96c84d6 SID:- _DVD-WQY46pfTXZ5RxAX3s UN: -
Authentication error cause: INCORRECT CREDENTIALS

3 б. Выдержка логов при успешной авторизации:

08:03:23.139 [WebContainer : 8] INFO c.b.g.c.u.m.i.UsernamePasswordCredentialsManagerImpl
RID:10c1bda5-3a2c-411b-a95e-64ff7d8e8b2a SID:- _DVD-WQY46pfTXZ5RxAX3s UN: - Channel =
IBS

08:03:23.139 [WebContainer : 8] INFO c.b.g.c.u.m.i.UsernamePasswordCredentialsManagerImpl
RID:10c1bda5-3a2c-411b-a95e-64ff7d8e8b2a SID:- _DVD-WQY46pfTXZ5RxAX3s UN: - Login =
tech-0100

08:03:23.166 [WebContainer : 8] INFO c.b.g.c.u.m.i.UsernamePasswordCredentialsManagerImpl
RID:10c1bda5-3a2c-411b-a95e-64ff7d8e8b2a SID:- _DVD-WQY46pfTXZ5RxAX3s UN: - Login OK

08:03:23.168 [hystrix-bcd.access.token.user-POST-access_tokenGroup-1] TRACE restServiceExternal
RID:10c1bda5-3a2c-411b-a95e-64ff7d8e8b2a SID:- _DVD-WQY46pfTXZ5RxAX3s UN: - REQUEST
- POST http://piphagor.abcmsec.ru:1410/abc-wire-mock/access_token

Headers: {Accept=[application/json], Content-Type=[application/x-www-form-urlencoded], Content-Length=[128]}

Body: client_id=bcd_online&client_secret=708f39dc-8a77-4558-90d5-317ef20a1df5&username=tech-0100&password=mx%25LxU&grant_type=password

3 с Скриншот и выдержка логов при отправке заявки на расчет персонального финансового плана:

Цели

☒ Машина

Стоимость цели: 1 111,00 В валюте: RUB

Дата реализации цели: 05.09.2021

☐ Кредит/займ
☐ Образование детей
☐ Пенсионный доход
☐ Платежи
☐ Накопление определенной суммы к определенному сроку
☐ Инвестиции
☐ Другие цели

Инвестирование

Валюта инвестирования: RUB

Сумма инвестирования: 1 111,00

Контактные данные

Внимание! Просто заполняя форму, Ваш финансовый советник свяжется с Вами для уточнения дополнительной информации и сообщит о времени готовности финансового плана.

Телефон: +79300000000

Выберите удобное для Вас время связи с финансовым советником.

Дата: 10.08.2021

Время: 09:30-10:00

Спасибо! Ваша заявка принята. В указанное время Ваш финансовый советник свяжется с Вами для уточнения информации и сообщит о времени готовности финансового плана.

08:20:26.067 [WebContainer : 2] INFO dbLogger RID:baa24eba-0dc0-4c06-994d-d435e03f4adf

SID:VLW0CKPmlgV-5qdx0-eea2D UN:tech-0100 - message UID=5b7369ac-9919-4a5d-9a60-

e2fdb9b68db, msg=<?xml version="1.0" encoding="UTF-8"?><soapenv:Body

xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

<client_service_request xmlns="http://www.mygemini.com/schemas/mygemini/"

xmlns:t="http://schemas.bcd.ru/is/clipboard/">

<system_block>

<t:originator>Gemini.WebPortal</t:originator>

<t:mean_for>IS.Clipboard</t:mean_for>

<t:document_id>9cafd6fa-248e-4607-9a37-c8b36646c116</t:document_id>

<t:event>New</t:event>

<t:event_time>2021-08-09T08:20:26.047+03:00</t:event_time>

</system_block>

<body>

<id>fb52c550-4038-48c1-945a-0c584a189e18</id>

<type>Ð—Ð°Ð°Ð°Ð· ÐŸÐ°ÐŸ</type>

<subject_id>f9a77de7-b830-426a-ba37-0964b4947d91</subject_id>


```

<ic>EMP011</ic>
<client_phone>+79500000000</client_phone>
<call_date>2021-08-10T09:00:00+03:00</call_date>
<investment_details>
  <sum>1111</sum>
  <sum_currency>RUR</sum_currency>
</investment_details>
<call_time_interval>60</call_time_interval>
<financial_goal_list>
  <financial_goal>
    <name>ÐœÐ°Ñ Ð½ÐµÐ½Ðµ</name>
    <description xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
    <sum>1111</sum>
    <sum_currency>RUR</sum_currency>
    <realization_date>2022-09-05+03:00</realization_date>
    <start_date xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
    <end_date xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
  </financial_goal>
</financial_goal_list>
</body>
</client_service_request>

```

Для выполнения заданий 4 и 5 потребуется подключение и исследование тестовых сайтов:

- <http://users.bugred.ru/> **Users** — телефонный справочник всех сотрудников. На любого из них можно повесить задачу (убрать квартиру, вынести мусор, поцеловать котика)
- <http://buggy.bugred.ru/> **Багги** — чистит данные для интернет магазинов. Убирает мусор из ФИО клиента + определяет по дате рождения категорию, чтобы отсеивать запрещенные для просмотра детьми товары. Чистить данные можно через файл или по одному человеку. В Багги есть дополнительная авторизация.
- **Доступ по SSH – один сервер для обоих сайтов.**

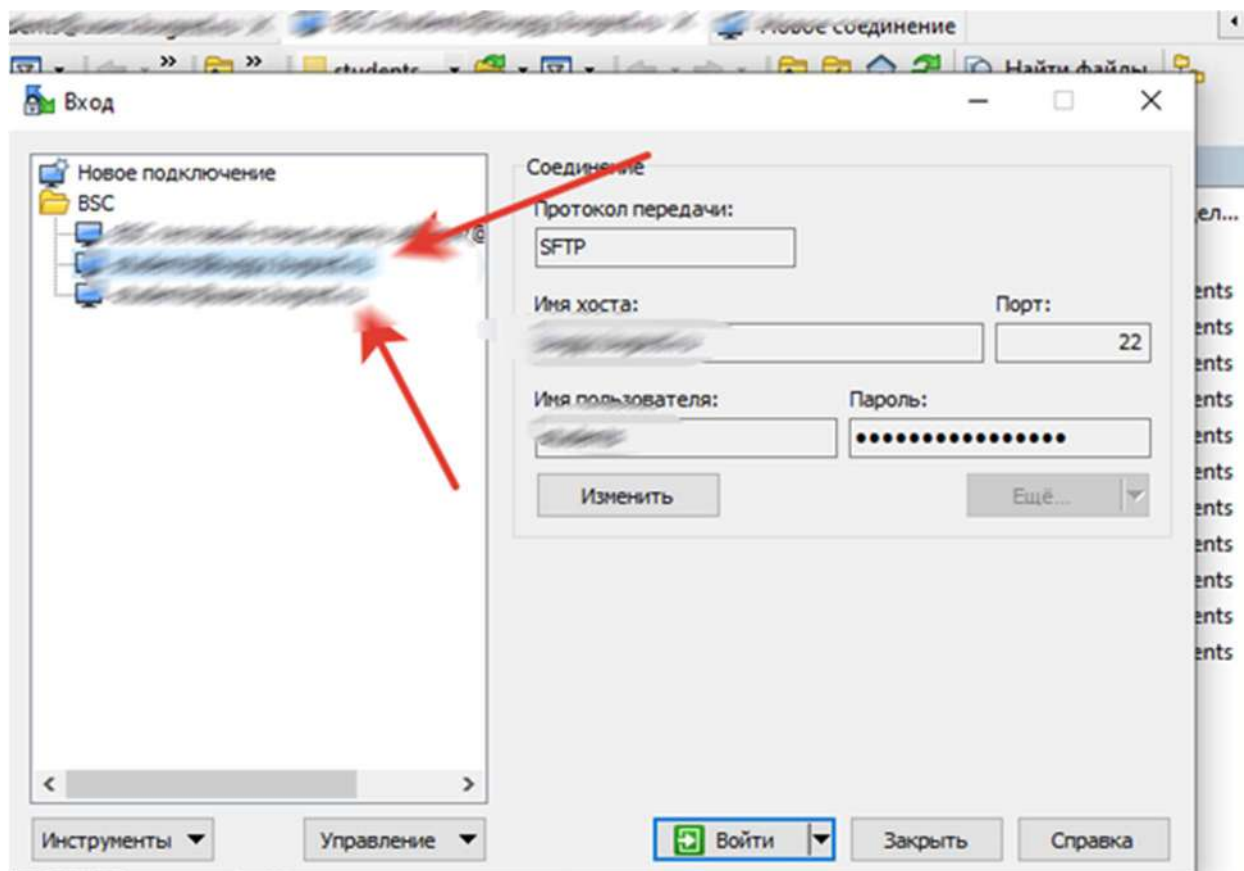
Задание 4. WinSCP и Putty

1. Познакомиться с инструментами (см. дополнительные материалы)
2. Установить обе программы
3. Настроить подключение к серверу сайтов Users и Buggy

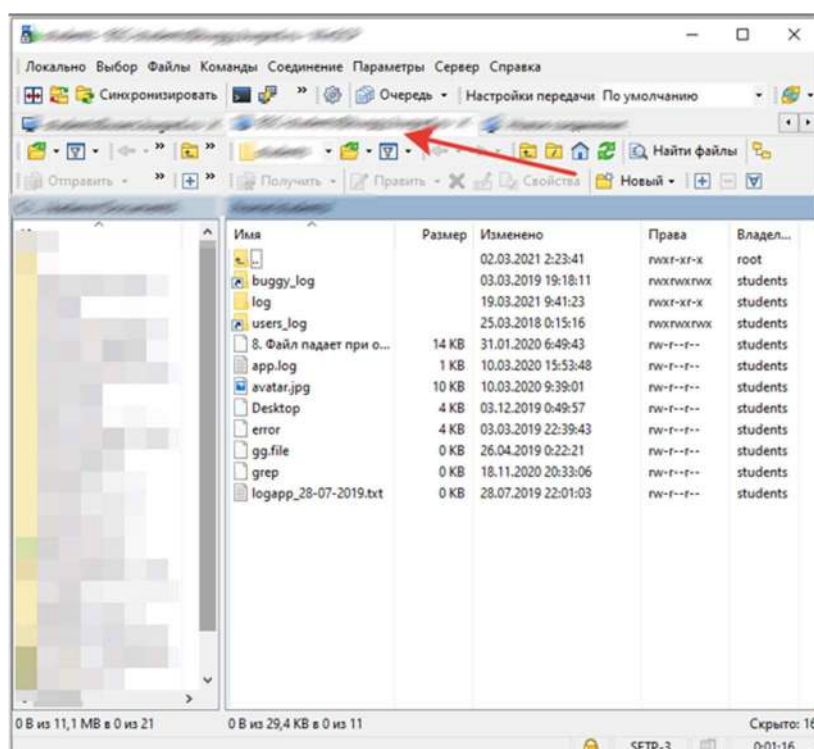
4. Сделать скриншоты подключений для отчета

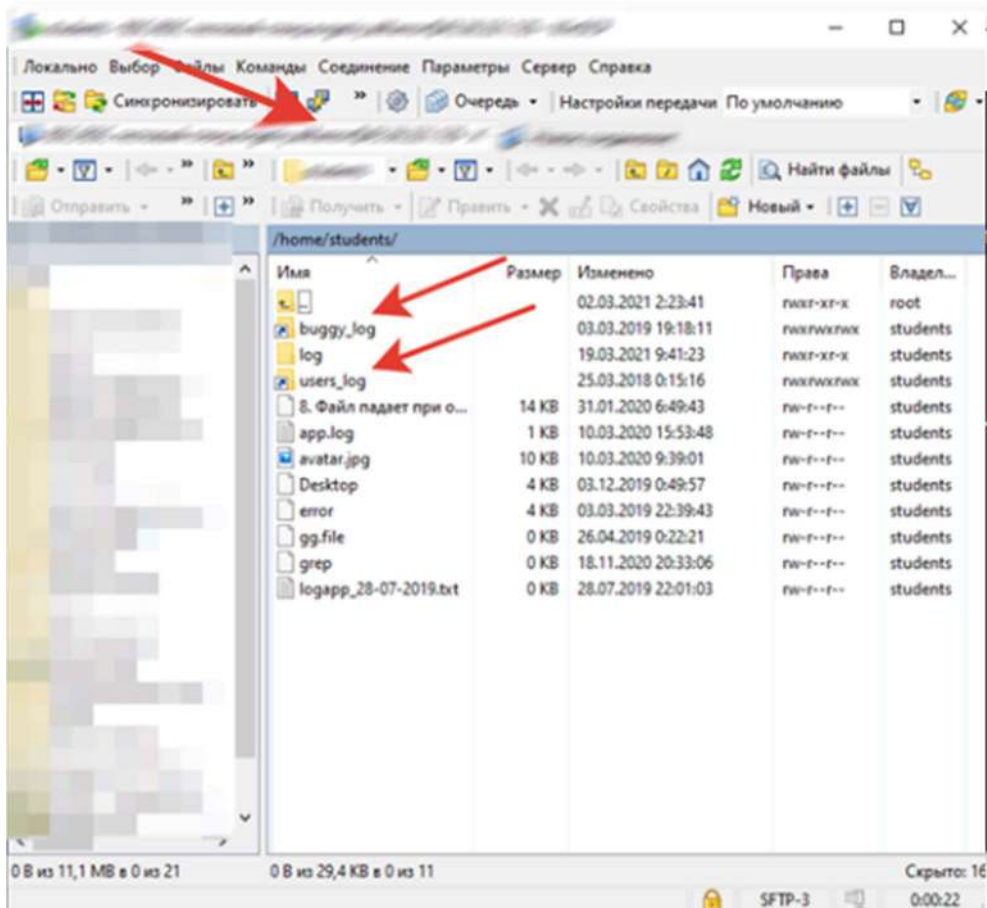
Решение:

Скриншот настроенных подключений WinCSP для сайтов <http://users.bugred.ru/> и <http://buggy.bugred.ru/>:

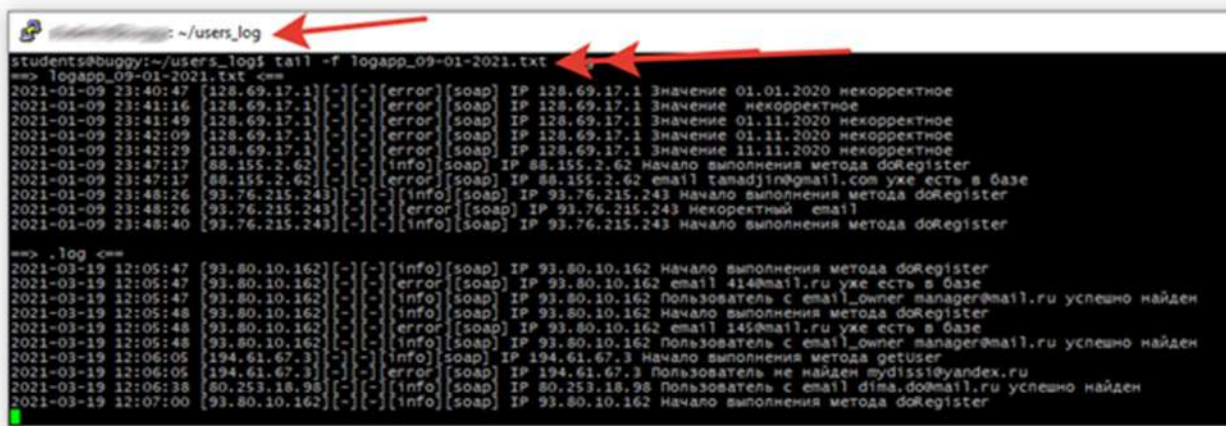


Скриншот активного подключения:





Так же и для Putty подключение с предустановками отображения:



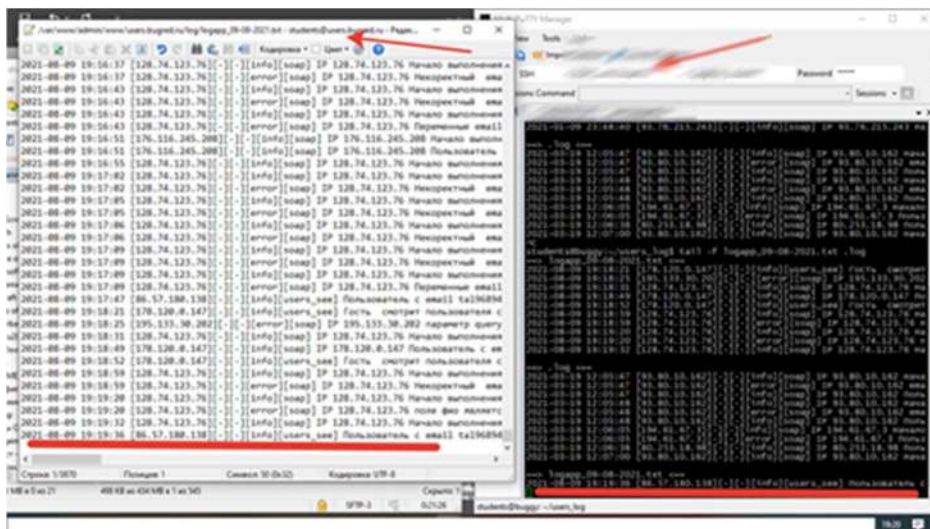
Задание 5. Users

1. Запустите WinSCP, найдите актуальный лог из Users (за сегодняшнее число), откройте. При необходимости поправьте кодировку
2. Войдите через Putty, начните следить за логом из Users (так, чтобы видеть в командной строке, когда там появляются новые записи).
3. Сделайте скриншоты обоих пунктов для отчета
4. Лог по операциям:

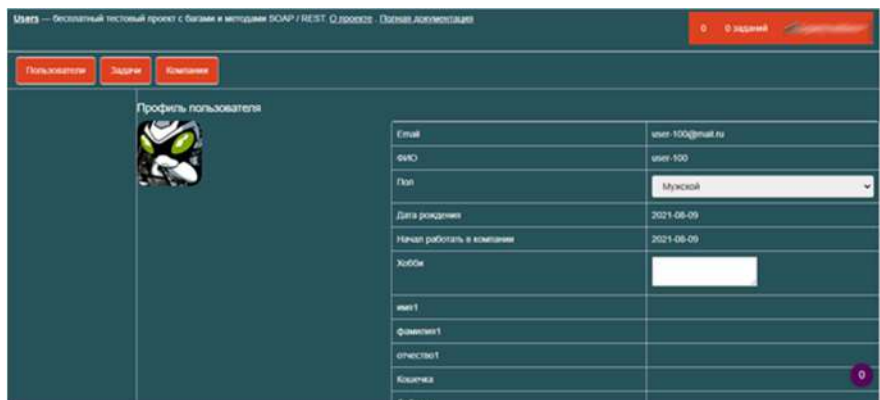
1. Выполнить следующий сценарий и в отчете показать **часть** лога, которая соответствует именно данному сценарию:
 1. Зарегистрироваться на Users (удобнее через ВК)
 2. Войти
 3. Добавить нового пользователя, у которого email **совпадает** с уже существующим пользователем
 4. Добавить пользователя с уникальным email, именем.
 5. Открыть данного пользователя на просмотр
- б. С помощью команды грейп достаньте из полученного лога:
 1. Все строки, в которых есть ошибки
 2. Все строки, в которых ошибок нет
 3. Так как мы отсекаем предупреждения, лучше ориентироваться как раз на уровень логирования.

Решение:

5.3 Скриншот одновременных, актуальных логов для WinSCP и Putty:



5.4.a.v открыл созданного пользователя с уникальным почтовым адресом на просмотр:



4 б С помощью команды грейп достаньте из полученного лога:

4. Все строки, в которых есть ошибки

Команда `grep "error" logapp_09-08-2021.txt`

5. Все строки, в которых ошибок нет

Команда `grep "info" logapp_09-08-2021.txt`

Поиск в рамках задания 5. 4. а можно поискать так:

Все строки, в которых есть ошибки

`grep "error" grep "user-100" logapp_09-08-2021.txt`

Все строки, в которых ошибок нет

`grep "info" | grep "user-100" logapp_09-08-2021.txt`

Пример вывода одной из выше описанных команд:

`students@buggy:~/users_log$ grep "info" | grep "user-100" logapp_09-08-2021.txt`

2021-08-09 19:54:41 [188.170.200.118][-][-][info][users_create] Пользователь с email user-100@mail.ru успешно создан

2021-08-09 19:55:25 [188.170.200.118][-][-][info][users_see] Пользователь с email смотрит пользователя с user-100@mail.ru

2021-08-09 20:06:05 [188.170.200.118][-][-][info][users_see] Пользователь с email смотрит пользователя с user-100@mail.ru

1. Зарегистрироваться на Users (удобнее через ВК)

2. Войти

3. Добавить нового пользователя, у которого email совпадает с уже существующим пользователем

(Искал ранее созданного пользователя tat2@tat.ru)

2021-08-10 17:58:32 [176.194.122.120][-][-][info][soap] IP 176.194.122.120 Начало выполнения метода doRegister

2021-08-10 17:58:32 [176.194.122.120][-][-][error][soap] IP 176.194.122.120 email tat2@tat.ru уже есть в базе

4. Добавить пользователя с уникальным email, именем.

2021-08-10 18:04:42 [188.170.200.118][-][-][info][users_create] Пользователь с email user-100@mail.ru успешно создан

5. Открыть данного пользователя на просмотр

2021-08-10 18:19:46 [188.170.200.118][-][-][info][users_see] Пользователь с email смотрит пользователя с user-100@mail.ru

Задание 6. Ответить на вопросы

Ответы:

1. Что можно узнать из логов?

Логи – это журнал действий системы. Чтобы в данном случае тестировщик всегда мог открыть этот файл и узнать: кто/что когда делал, во сколько вышел/зашел, какая информация передавалась от одного компонента к другому, какой ответ пришел от сторонней системы. Вся эта информация протоколируется, так легче локализовать проблему и “найти концы”.

- Зачем нужны логи тестировщику?

Потребность чтения логов у тестировщика возникает при анализе инцидентов, багов от заказчика. От тестировщика требуется умение логи «прочитать», понять последовательность шагов и в итоге локализовать проблему. Логи являются неотъемлемой частью инструментов тестирования бекенда, интеграции, сервисов. Как один сервис передает запрос по цепочке в другой, все это возможно понять из логов.

2. Уровни логирования и их отличия:

► **Debug**: сообщения отладки, профилирования. Debug — журналирование моментов вызова «крупных» операций. Старт/остановка потока, запрос пользователя и т.п.

► **Trace** — вывод всего подряд. Если Debug не позволяет локализовать ошибку. В нем полезно отмечать вызовы разнообразных блокирующих и асинхронных операций.

► **Info**: обычные сообщения, информирующие о действиях системы. Это разовые операции, которые повторяются крайне редко, но регулярно. Вся стандартная информация идет с таким уровнем.

► **Warn**: возможно произошла ошибка, это новый тип ситуации, ещё не известный системе. Следует разобраться в том, что произошло, что это означает, и отнести ситуацию либо к инфо-сообщению, либо к ошибке. Соответственно, придётся доработать код обработки таких ситуаций.

► **Error**: ошибка в работе системы, требующая вмешательства. Что-то не сохранилось, что-то отвалилось. Необходимо срочно принимать меры. Ошибки этого уровня и выше требуют немедленной записи в лог, чтобы ускорить реакцию на них. Но если пользователь ввёл в поле -1, где это не предполагалось – не надо писать об этом в лог ошибок.

► **Fatal**: этот класс ошибок, относящийся к разряду критических. Такие ошибки приводят к неработоспособности системы, подсистем. Чаще всего случаются фатальные ошибки из-за

неверной конфигурации или отказов оборудования. Требуют срочной реакции и лучше на них настроить немедленные уведомления.

3. Для чего нужна команда `grep`?

Команда `grep` (печать глобального регулярного выражения) это команда в *Linux* системах. `grep` ищет в одном или нескольких входных файлах строки, соответствующие заданному шаблону, и записывает каждую соответствующую строку в стандартный вывод.

Синтаксис команды `grep` следующий (пункты в квадратных скобках необязательны):

`grep [OPTIONS] PATTERN [FILE...]`

OPTIONS — ноль или более вариантов. `Grep` включает ряд опций, управляющих его поведением.

PATTERN — шаблон поиска.

FILE — ноль или более имен входных файлов.

4. Перечислить `cd` по порядку команды, которые необходимо сделать, чтобы запустить лог на непрерывный просмотр в `putty`

Команды `cd`(мы в корневой директории):

`cd users_log`

Далее идет команда `tail`:

`tail -f logapp_09-08-2021.txt`

6. API

Для любого запроса провести тест-дизайн, описать чеклист проверок на api метода (в любом удобном виде). Сформировать коллекцию запросов с тестами на выбранный метод.

Тест дизайн

Коллекция запросов

Легенда: **Красный цвет** — негативный сценарий, **бирюзовый** — тип параметра

Размещение заказа на нового питомца POST <https://petstore.swagger.io/v2/store/order>

id (целочисленный):

нет параметра

пусто

содержит символы

буквы

2

-1

0

10000000000000000

petId (целочисленный)

нет параметра

пусто

содержит символы

буквы

2

-1

0

10000000000000000

quantity (целочисленный)

нет параметра

пусто

содержит символы

буквы

2

-1

0

10000000000000000

shipDate (строчный date-time)

нет параметра

пусто

содержит символы

буквы

2021-08-13T16:45:45.121Z

status (строчный)

нет параметра

пусто

содержит символы

цифры

approved

буквы

complete (boolean)

нет параметра

пусто

содержит символы

буквы

true

false

Чек-лист проверок

Кейс №1 «Корректные параметры запроса»

Цель: проверка успешной генерации и отправки ссылки

| № | Шаг | Ожидаемый результат |
|---|--|---|
| 1 | Сформирован запрос на генерацию и отправку ссылки https://petstore.swagger.io/v2/store/order Метод: POST. В теле запроса следующие параметры заполнены валидными данными, например: <pre>{ "id": 2, "petId": 12, "quantity": 0, "shipDate": "2021-08-13T16:45:45.121Z", "status": "placed", "complete": true }</pre> | Успешная генерация и отправка ссылки Code response: 200 Тело ответа содержит: <pre>{ "id": 2, "petId": 12, "quantity": 0, "shipDate": "2021-08-13T16:45:45.121+0000", "status": "placed", "complete": true }</pre> |
| 2 | Сформирован запрос на генерацию и отправку ссылки https://petstore.swagger.io/v2/store/order Метод: POST. В теле запроса следующие параметры заполнены валидными данными, в поле id поставить 10000000000000000: <pre>{ "id": 10000000000000000, "petId": 12, "quantity": 0, "shipDate": "2021-08-13T16:45:45.121Z", "status": "placed", "complete": true }</pre> | Тело ответа содержит: <pre>{ "id": 10000000000000000, "petId": 12, "quantity": 0, "shipDate": "2021-08-13T16:45:45.121+0000", "status": "placed", "complete": true }</pre> |

Кейс №2 «Некорректные параметры запроса»

Цель: проверка на некорректные значения

| | | |
|---|---|---|
| 1 | <p>Сформирован запрос на генерацию и отправку ссылки https://petstore.swagger.io/v2/store/order</p> <p>Метод: POST. В теле запроса следующие параметры заполнены валидными данными, кроме id. Данное поле пустое:</p> <pre>{ "id": , "petId": 12, "quantity": 0, "shipDate": "2021-08-13T16:45:45.121Z", "status": "placed", "complete": true }</pre> | <p>Код ошибки:</p> <p>Code response: 400 Bad request</p> <p>Тело ответа содержит:</p> <pre>{ "code": 400, "type": "unknown", "message": "bad input" }</pre> |
| 2 | <p>Сформирован запрос на генерацию и отправку ссылки https://petstore.swagger.io/v2/store/order</p> <p>Метод: POST. В теле запроса следующие параметры заполнены валидными данными, кроме id. Данное поле отсутствует:</p> <pre>{ "petId": 12, "quantity": 0, "shipDate": "2021-08-13T16:45:45.121Z", "status": "placed", "complete": true }</pre> | <p>Код ошибки:</p> <p>Code response: 400 Bad request</p> <p>Тело ответа содержит:</p> <pre>{ "code": 400, "type": "unknown", "message": "bad input" }</pre> |
| 3 | <p>Сформирован запрос на генерацию и отправку ссылки https://petstore.swagger.io/v2/store/order</p> <p>Метод: POST. В теле запроса следующие параметры заполнены валидными данными, кроме id. В данном поле присутствуют символы:</p> <pre>{ "id": #, "petId": 12,</pre> | <p>Код ошибки:</p> <p>Code response: 400 Bad request</p> <p>Тело ответа содержит:</p> <pre>{ "code": 400, "type": "unknown", "message": "bad input" }</pre> |

| | | |
|---|--|---|
| | <pre>"quantity": 0, "shipDate": "2021-08-13T16:45:45.121Z", "status": "placed", "complete": true }</pre> | <pre>}</pre> |
| 4 | <p>Сформирован запрос на генерацию и отправку ссылки https://petstore.swagger.io/v2/store/order</p> <p>Метод: POST. В теле запроса следующие параметры заполнены валидными данными, кроме id. В данном поле присутствуют буквы:</p> <pre>{ "id": "AaBb", "petId": 12, "quantity": 0, "shipDate": "2021-08-13T16:45:45.121Z", "status": "placed", "complete": true }</pre> | <p>Код ошибки:</p> <p>Code response: 400 Bad request</p> <p>Тело ответа содержит:</p> <pre>{ "code": 400, "type": "unknown", "message": "bad input" }</pre> |
| 5 | <p>Сформирован запрос на генерацию и отправку ссылки https://petstore.swagger.io/v2/store/order</p> <p>Метод: POST. В теле запроса следующие параметры заполнены валидными данными, кроме id. В данном поле отрицательное значение:</p> <pre>{ "id": -1, "petId": 12, "quantity": 0, "shipDate": "2021-08-13T16:45:45.121Z", "status": "placed", "complete": true }</pre> | <p>Тело ответа содержит:</p> <pre>{ "code": 400, "type": "unknown", "message": "bad input" }</pre> |

Цель: дополнительная проверка на неоднозначные значения

| | | |
|---|---|---|
| 1 | <p>Сформирован запрос на генерацию и отправку ссылки https://petstore.swagger.io/v2/store/order Метод: POST. В теле запроса следующие параметры заполнены валидными данными, в id поставить 0:</p> <pre>{ "id": 0, "petId": 12, "quantity": 0, "shipDate": "2021-08-13T16:45:45.121Z", "status": "placed", "complete": true }</pre> | <p>Код ошибки: Code response: 400 Bad request Тело ответа содержит: <pre>{ "code": 400, "type": "unknown", "message": "bad input" }</pre></p> |
|---|---|---|

7. DevTools

2. Перейдите по ссылке. Корректно ли отображается страница для экрана мобильного телефона?

Данная страница отображается так же, как и для версии ПК, страница не адаптивна под мобильные устройства. Что бы ответить корректна отображается или нет нужно знать предусматривается ли адаптивность под мобильные разрешения на этом сайте в принципе. И если не предусматривалось, то страница отображается корректно.

1. Авторизоваться на портале. С помощью DevTools найдите cookie с названием _bcdpks. В качестве ответа в поле ввода напишите размер этой cookie (только число):

35

2. Перейдите по ссылке. Корректно ли отображается страница для экрана мобильного телефона?

Нет

3. Перейдите по ссылке. Выберите в блоке "Вклады и инвестиции" продукт ИСЖ и перейдите на страницу деталей. Найдите запрос, в рамках которого были получены данные для отображения деталей продукта. В качестве ответа в поле ввода напишите метод этого запроса:

POST

4. Перейдите по ссылке. Выберите в блоке "Вклады и инвестиции" продукт ИСЖ и перейдите на страницу деталей. В качестве ответа в поле ввода укажите декодированное значение заголовка Authorization, который был отправлен в запросе:

Basic cWE6MXEYdzNlNHl1dDZ5

5. Перейдите по ссылк. Выберите в блоке "Вклады и инвестиции" продукт ИСЖ и перейдите на страницу деталей. В качестве ответа укажите значение id элемента кнопки (button) "Изменить" на форме, которая открывается при редактировании названия продукта

2

6. Перейдите по ссылке. Далее перейдите на форму Продукты > Список счетов. В качестве ответа в поле ввода напишите значение параметра productTypes, который был передан в теле запроса POST /product/getProducts

GA_RUS

7. Перейдите по ссылке <http://test.a-b-c.ru/bcd3/>. В качестве ответа в поле ввода напишите номер строки скрипта, при выполнении которой произошла ошибка ERR_NAME_NOT_RESOLVED

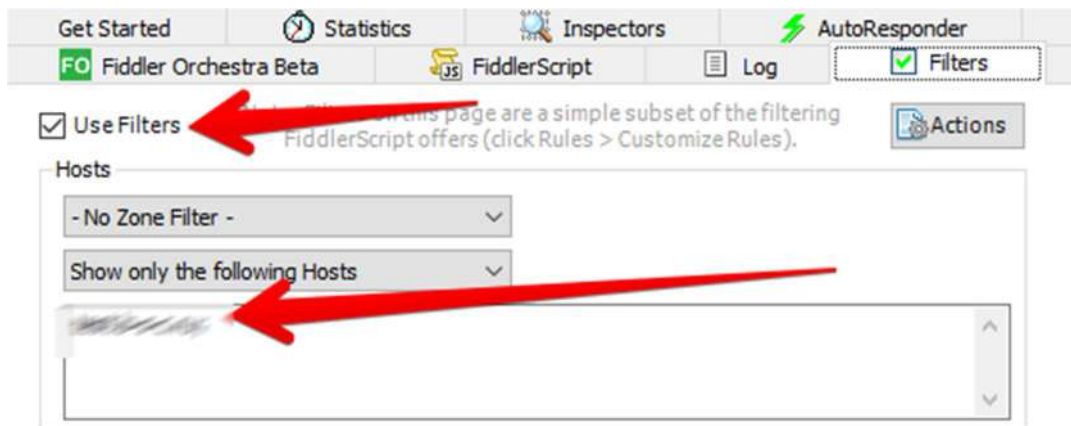
6 строка в Elements

8. Перейдите по ссылке <http://testingchallenges.thetestingmap.org/challenge5.php>. При нажатии на кнопку Submit отправляется запрос для определения типа браузера и ip клиента. Вам нужно определить URL, куда отправляется запрос, и ввести его в поле ввода в качестве ответа:

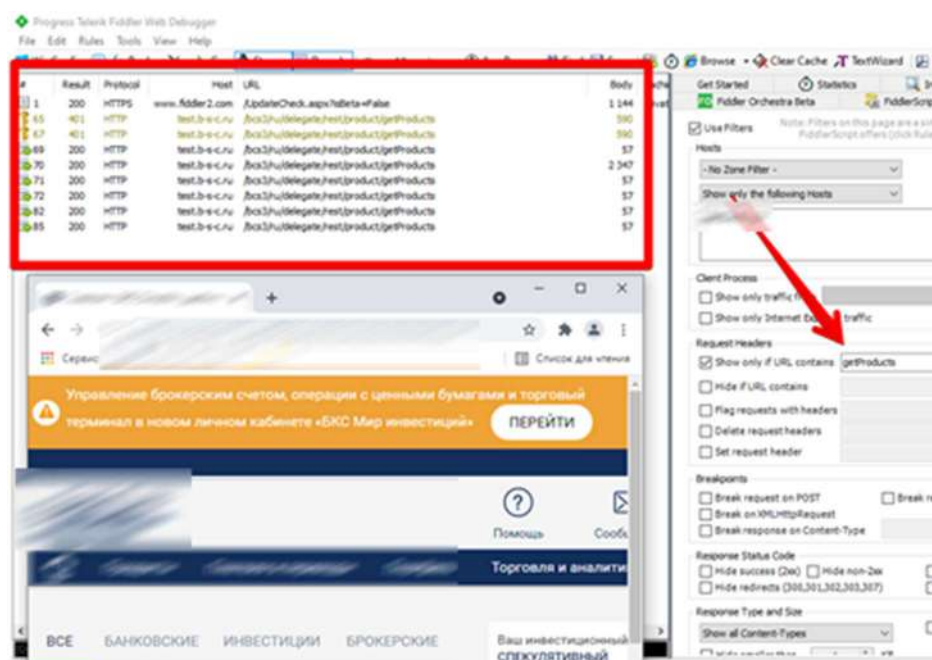
http://webanalyticsengine.thetestingmap.org/web_analytics_engine.php

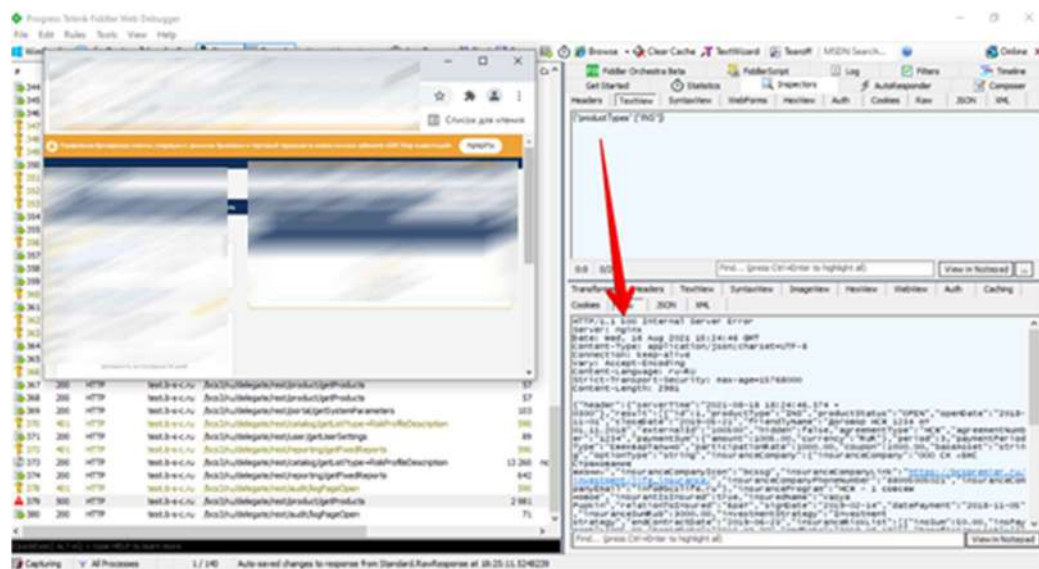
8. Инструменты тестирования (мои ответы в скриншотах).

1. Настроить перехват трафика через fiddler для компьютера и мобильного устройства (см. приложенный мануал)
 - a. Настроить фильтр так, чтобы выводился трафик только для сервера.



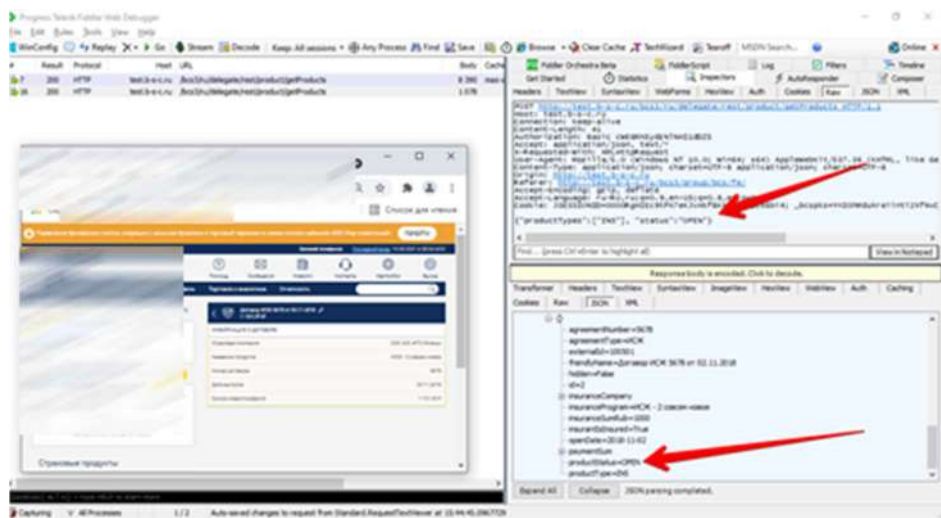
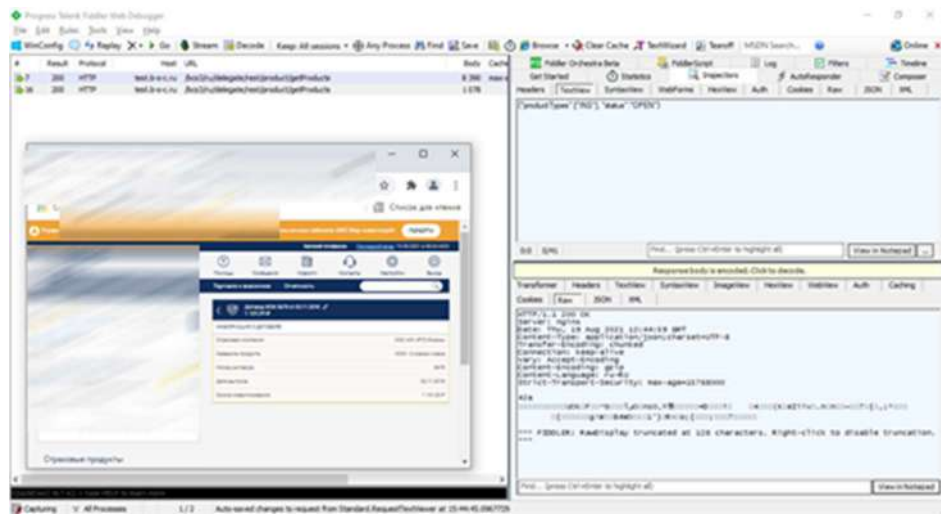
2. Использование фильтрации и подмены ответа от сервера (работаем с главной страницей портала - Продукты)
 - a. Применить фильтр для запросов (вкладка Filters) таким образом, чтобы в списке остались только запросы получения данных продуктов (getProducts) при переходе на главную страницу ABC Онлайн
 - b. Перехватить ответ для запроса получения данных продукта “Инвестиционное страхование жизни” (ProductTypes = INS) и изменить его http статус на 500.
 - c. В качестве ответа приложить скриншот Fiddler, на котором видно установленный фильтр, список запросов, а также скриншот главной страницы ABC Онлайн из браузера и мобильного устройства





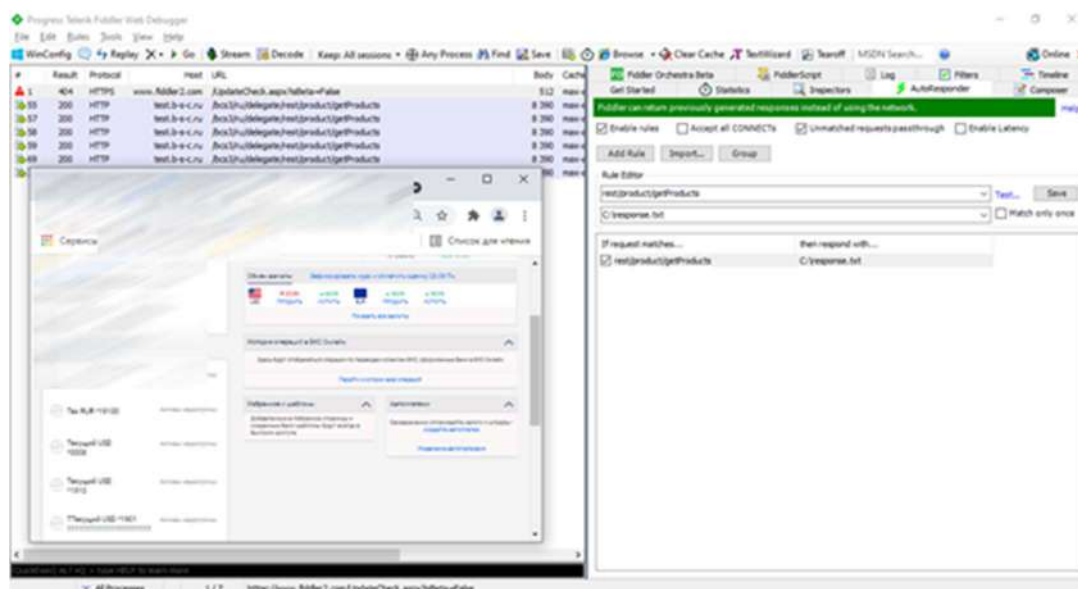
3. Использование подмены запроса с фронта

- Перехватить запрос получения данных продукта “Инвестиционное страхование жизни” (ProductTypes = INS) и изменить значение параметра status на значение CLOSE.
- В качестве ответа приложить скриншот Fiddler, на котором видно тело отправленного запроса и полностью тело полученного ответа, а также скриншот главной страницы ABC Онлайн **из браузера и с мобильного устройства.**

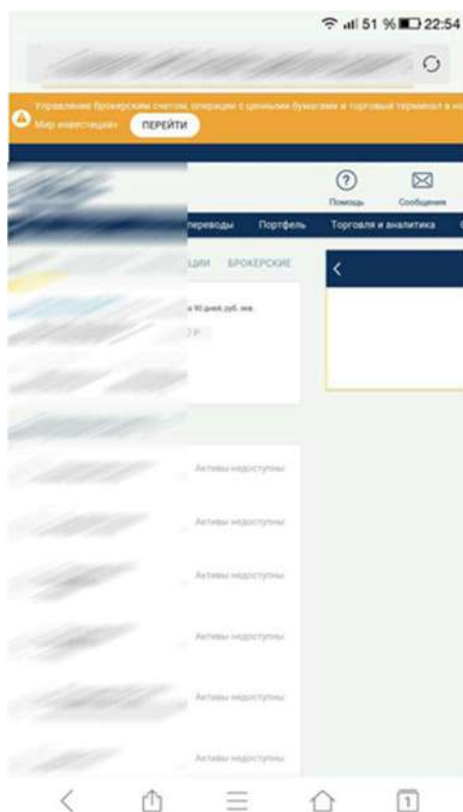


4. Использование AutoResponder

- a. В рамках выполнения задания нужно на вкладке AutoResponder создать новое правило, которое бы соответствовало запросу `rest/product/getProducts` только для продуктов типа А.
- b. Добавить в качестве ответа успешный ответ с телом `response.txt`
- c. Активировать правило и авторизоваться на портале.
- d. В качестве ответа приложить скриншот активной вкладки AutoResponder, а также скриншот главной страницы.



Дополнительно с телефона:



9. Приложение 1. Методы проектирования тестов на основе опыта.



Методы
проектирования тс

10. Приложение 2. Изучение продукта и интеллект-карты.



Изучение
продукта и интелле

11. Приложение 3. Попарное тестирование.



Попарное
тестирование.xlsx

12. Дополнительный материал

➤ Виды тестирования:



Виды
тестирования.pdf

- ISTQB - Сертификационный совет по тестированию программного обеспечения, действующий на международном уровне. (<https://www.rstqb.org/ru/istqb-downloads.html>)
- Карта Формула успеха тестировщика:



карта Формула
успеха тестировщи