МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«БГТУ им. В.Г.ШУХОВА»

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

КУРСОВАЯ РАБОТА

по дисциплине:

«Тестирование программных систем»

тема: интеграционное тестирование Веб-приложения

Автор работы _____Колесников Владимир Дмитриевич, ПВ-41

(подпись)

Руководитель проекта_____ Поляков Владимир Константинович

(подпись)

Оценка _____

Белгород
2020 г.

Оглавление

Введение

Все современные достижения человечества делают нашу жизнь максимально простой. И сегодня все можно выполнить буквально одним нажатием клавиши. Но чтобы достичь подобного результата, необходимо создать максимально качественную и работоспособную логику, которая могла бы функционировать даже в самые критические моменты.

Конечно, со временем, созданные технологии могут функционировать не так корректно, как предполагалось изначально. Да и итоги применения того или иного программного обеспечения могут не соответствовать заявленным требованиям. Все это доказывает тот факт, что процесс тестирования любого программного обеспечения играет крайне важную роль.

Постановка задачи

Цель курсовой работы – интеграционное тестирование веб-приложения а также изучение возможностей различных средств и приложений для упрощения интеграционного тестирования.

Теоретические сведения

Чтобы разобраться в интеграционном тестировании, сначала надо определить понятие традиционного процесса проверки программного обеспечения. Итак, тестирование ПО — это специфическая деятельность, которая проверяет, соблюдаются ли заявленные требования в текущей работе веб-продукта.

Другими словами, фактический результат работы ПО должен соответствовать ожидаемому результату. Иначе в запрограммированную структуру программного обеспечения придется вносить коррективы или переписывать код заново.

Проверка работоспособности программного обеспечения проводиться на разных уровнях. Базовыми среди них являются:

- Модульное тестирование
- Интеграционное тестирование
- Системное тестирование
- Приемочное тестирование

Основной интерес в данной работе представляет интеграционное тестирование.

Глава 1. Интеграционное тестирование

Интеграционное тестирование — вид тестирования, при котором на соответствие требований проверяется интеграция модулей, их взаимодействие между собой, а также интеграция подсистем в одну общую

систему. Для интеграционного тестирования используются компоненты, уже проверенные с помощью модульного тестирования, которые группируются в множества. Данные множества проверяются в соответствии с планом тестирования, составленным для них, а объединяются они через свои интерфейсы.

Так как модули соединяются между собой с помощью предусмотренных реализацией интерфейсов и в процессе тестирования, нет необходимости рассматривать внутреннюю структуру компонентов, и можно утверждать, что интеграционное тестирование выполняется методом «черного ящика».

Интеграционное тестирование в качестве входных данных использует модули, над которыми было проведено модульное тестирование, группирует их в более крупные множества, выполняет тесты, определённые в плане тестирования для этих множеств, и представляет их в качестве выходных данных и входных для последующего системного тестирования.

Целью интеграционного тестирования является проверка соответствия проектируемых единиц функциональным, приёмным и требованиям надежности. Тестирование этих проектируемых единиц — объединения, множества или группы модулей — выполняется через их интерфейс, с использованием тестирования «чёрного ящика».

Одним из наиболее распространённых средств интеграционного тестирования веб-сервисов является Selenium WebDriver.

Глава 2. Selenium WebDriver

Selenium WebDriver — это инструмент для автоматизации действий веб-браузера. В большинстве случаев используется для тестирования веб-приложений, но этим не ограничивается. В частности, он может быть использован для решения рутинных задач администрирования сайта или регулярного получения данных из различных источников (сайтов).

По назначению Selenium WebDriver представляет собой драйвер браузера, то есть программную библиотеку, которая позволяет разрабатывать программы, управляющие поведением браузера.

По своей сущности Selenium WebDriver представляет собой:

- Спецификацию программного интерфейса для управления браузером,
- Референсные реализации этого интерфейса для нескольких браузеров,
- Набор клиентских библиотек для этого интерфейса на нескольких языках программирования.

Selenium WebDriver — это в первую очередь набор библиотек для различных языков программирования. Эти библиотеки используются для отправки HTTP-запросов драйверу с помощью протокола *JsonWireProtocol*. В

данных запросах указано действие, которое должен совершить браузер в рамках текущей сессии. Примерами таких команд могут быть команды нахождения элементов по локатору, переход по ссылкам, парсинг текста страницы или элемента, нажатие кнопок или переход по ссылкам на странице веб-сайта.

Глава 3. Процесс создания тестов

Для работы с Selenium WebDriver необходимо 3 основных программных компонента:

1. Браузер, работу которого пользователь хочет автоматизировать. Это реальный браузер определенной версии, установленный на определенной ОС и имеющий свои настройки. В данной работе будет использоваться браузер Chrome.
2. Для управления браузером необходим драйвер браузера. Драйвер в данном случае на самом деле является веб сервером, который запускает браузер и отправляет ему команды, а также закрывает его. У каждого браузера свой драйвер. Связано это с тем, что у каждого браузера свои отличные команды управления и реализованы они по-своему. Найти список доступных драйверов и ссылки для скачивания можно на официальном сайте Selenium проекта. Поскольку использоваться в данной работе будет браузер Chrome, то драйвер, соответственно, подбирается для него.
3. Скрипт/тест, который содержит набор команд на определенном языке программирования для драйвера браузера. Такие скрипты используют Selenium Webdriver bindings (готовые библиотеки), которые доступны пользователям на различных языках. В данной работе все тесты будут написаны на языке Java. Загрузка необходимых библиотек будет осуществляться при помощи Maven.

Заключение

В ходе выполнения курсовой работы были получены навыки в интеграционном тестировании, закреплены навыки тестирования с помощью библиотеки junit, получены навыки использования Selenium WebDriver.

Список использованной литературы

1. **Интеграционное тестирование.** [Электронный ресурс] – Режим доступа: https://qalight.com.ua/baza-znaniy/integratsionnoe-testirovanie
2. **Интеграционное тестирование. Основные понятия, характерные особенности и примеры.** [Электронный ресурс] – Режим доступа: https://testmatick.com/ru/integratsionnoe-testirovanie-osnovnye-ponyatiya-harakternye-osobennosti-i-primery

3. **Selenium** [Электронный ресурс] – Режим доступа: https://www.selenium.dev
4. **WebDriver. Обзор и принцип работы.** [Электронный ресурс] – Режим доступа: https://kreisfahrer.gitbooks.io/selenium-webdriver/content/webdriver_intro/webdriver_obzor_i_printsip_raboti.html
5. **Selenium - Википедия** [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Selenium

Приложение

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.example</groupId>
    <artifactId>Course_Testing</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>15</maven.compiler.source>
        <maven.compiler.target>15</maven.compiler.target>
    </properties>

    <dependencies>
        <dependency>
            <groupId>org.seleniumhq.selenium</groupId>
            <artifactId>selenium-java</artifactId>
            <version>3.141.59</version>
        </dependency>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.13</version>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>commons-io</groupId>
            <artifactId>commons-io</artifactId>
            <version>2.5</version>
        </dependency>
    </dependencies>

</project>
```

config.properties

```
chromedriver = src/drivers/chromedriver.exe

wowhead.mainpage = https://www.wowhead.com/
wowhead.items = https://www.wowhead.com/items/
```

ConfigPropertiesReader

```
package okayga;

import java.io.FileInputStream;
import java.io.IOException;
import java.util.Properties;

public class ConfigPropertiesReader {
    protected static FileInputStream fileInputStream;
    protected static Properties PROPERTIES;
    static {
        try {
            //указание пути до файла с настройками
            fileInputStream = new
FileInputStream("src/test/resources/config.properties");
            PROPERTIES = new Properties();
            PROPERTIES.load(fileInputStream);
        } catch (IOException e) {
            e.printStackTrace();
            //обработка возможного исключения (нет файла и т.п.)
        } finally {
            if (fileInputStream != null)
                try {
                    fileInputStream.close();
                } catch (IOException e) {
                    e.printStackTrace(); } } }
    public static String getProperty(String key) {
        return PROPERTIES.getProperty(key);
    }
}
```

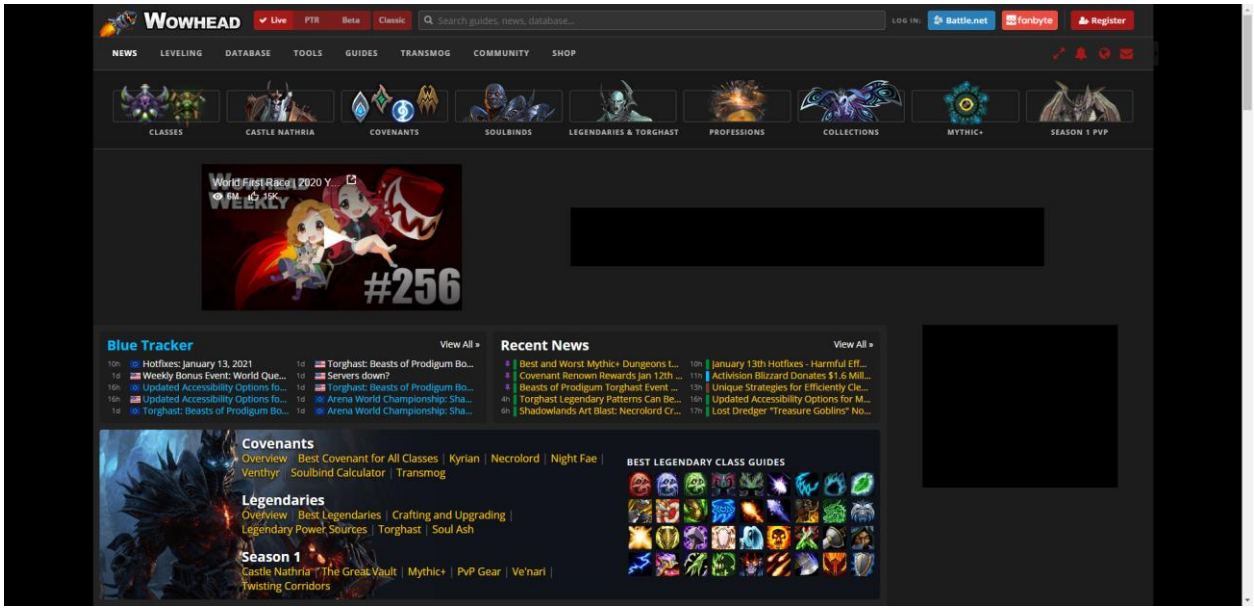
WowheadWebTest

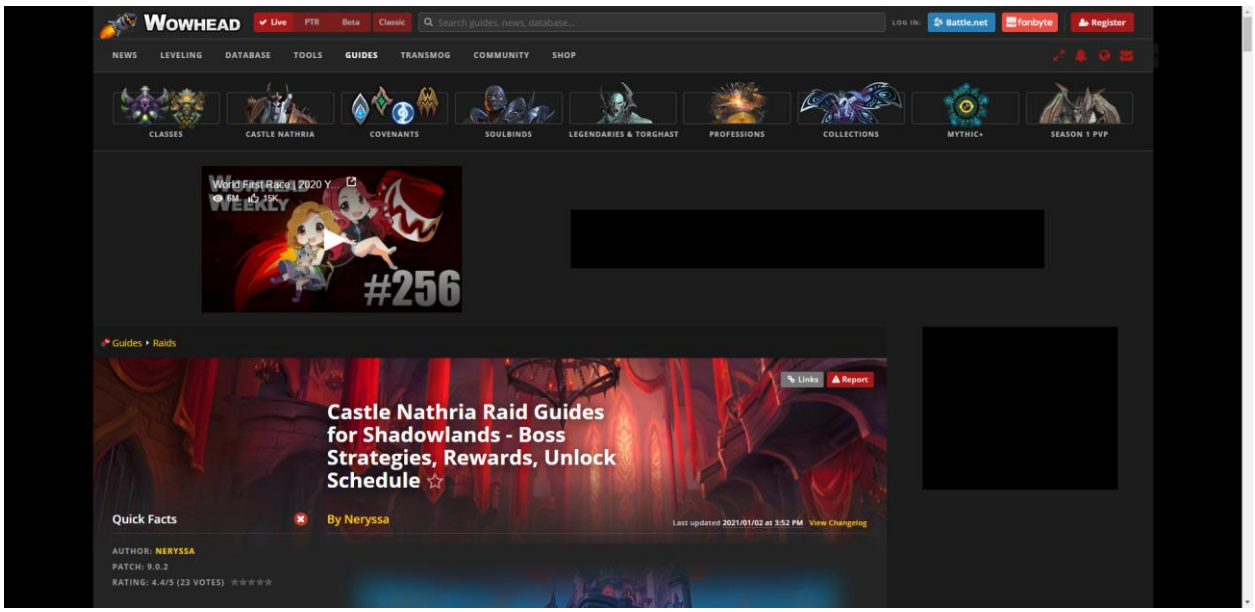
```
package okayga;

import org.apache.commons.io.FileUtils;
import org.junit.AfterClass;
import org.junit.BeforeClass;
import org.junit.Test;
import org.openqa.selenium.*;
import org.openqa.selenium.support.ui.Select;
import org.openqa.selenium.chrome.ChromeDriver;

import java.io.File;
import java.io.IOException;
import java.util.List;
import java.util.concurrent.TimeUnit;

import static org.junit.Assert.*;

public class WowheadWebTest{
    public static WebDriver driver;

    @BeforeClass
    public static void setup(){
        System.out.println("Testing Wowhead.");
        System.setProperty("webdriver.chrome.driver",
ConfigPropertiesReader.getProperty("chromedriver"));
        driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
```

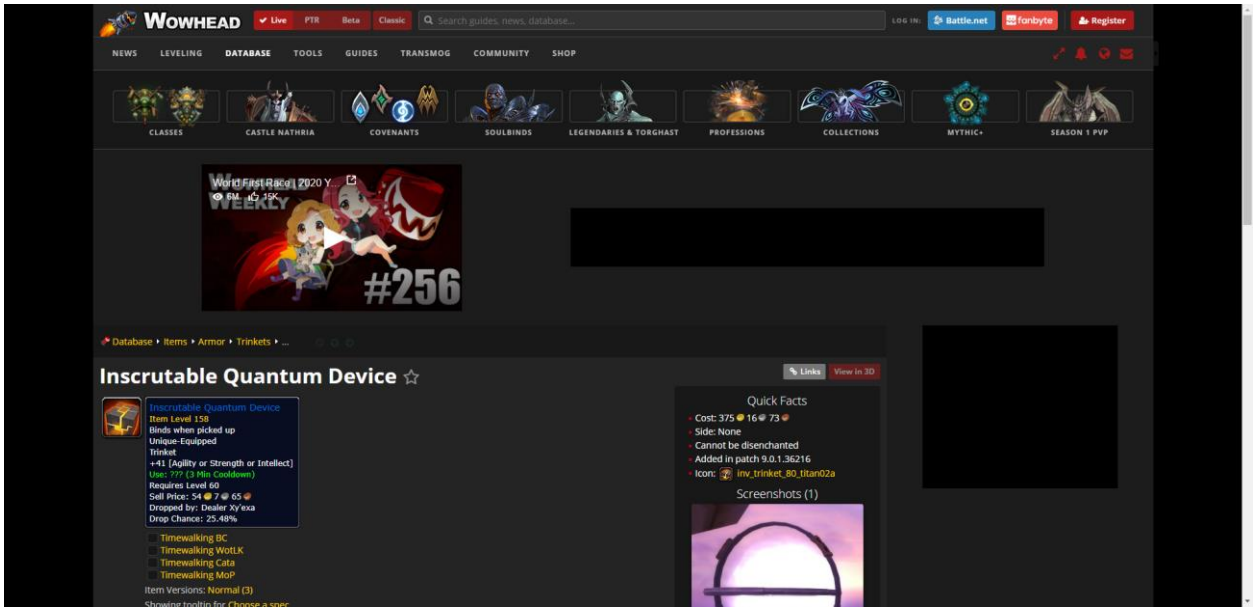
```
    }

    @Test
    public void TestMainPage(){
        driver.get(ConfigPropertiesReader.getProperty("wowhead.mainpage"));
        String url = driver.getCurrentUrl();
        try {
            File scrFile =
((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
            FileUtils.copyFile(scrFile, new
File(".\\src\\screenshots\\MainPage.png"));
        } catch (IOException e) {
            System.out.println("Failed to capture screenshot: " + e.getMessage());
        }
        assertEquals("https://www.wowhead.com/",url);
    }

    @Test
    public void TestClickEntity(){
        driver.get(ConfigPropertiesReader.getProperty("wowhead.mainpage"));
        driver.findElement(By.className("header-nav-sire-denathrius")).click();
        String title = driver.findElement(By.className("title")).getText();
        try {
            File scrFile =
((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
            FileUtils.copyFile(scrFile, new File(".\\src\\screenshots\\Entity.png"));
        } catch (IOException e) {
            System.out.println("Failed to capture screenshot: " + e.getMessage());
        }
        assertEquals("Castle Nathria Raid Guides",title);
    }

    @Test
    public void TestClickFirstItem() {
        driver.get(ConfigPropertiesReader.getProperty("wowhead.items"));
        driver.findElements(By.className("listview-row")).get(0).click();
        String name = driver.findElement(By.className("heading-size-1")).getText();
        try {
            File scrFile =
((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
            FileUtils.copyFile(scrFile, new
File(".\\src\\screenshots\\FirstItem.png"));
        } catch (IOException e) {
            System.out.println("Failed to capture screenshot: " + e.getMessage());
        }
        assertNotNull(name);
    }

    @Test
    public void TestCompareTwoItems() {
        driver.get(ConfigPropertiesReader.getProperty("wowhead.items"));
        List<WebElement> items = driver.findElements(By.className("listview-row"));
        String name1 = items.get(1).findElement(By.className("listview-
cleartext")).getText();
        String name2 = items.get(2).findElement(By.className("listview-
cleartext")).getText();
        assertNotEquals(name1, name2);
    }

    @Test
    public void ClickFirstItemWithFilters() {
```

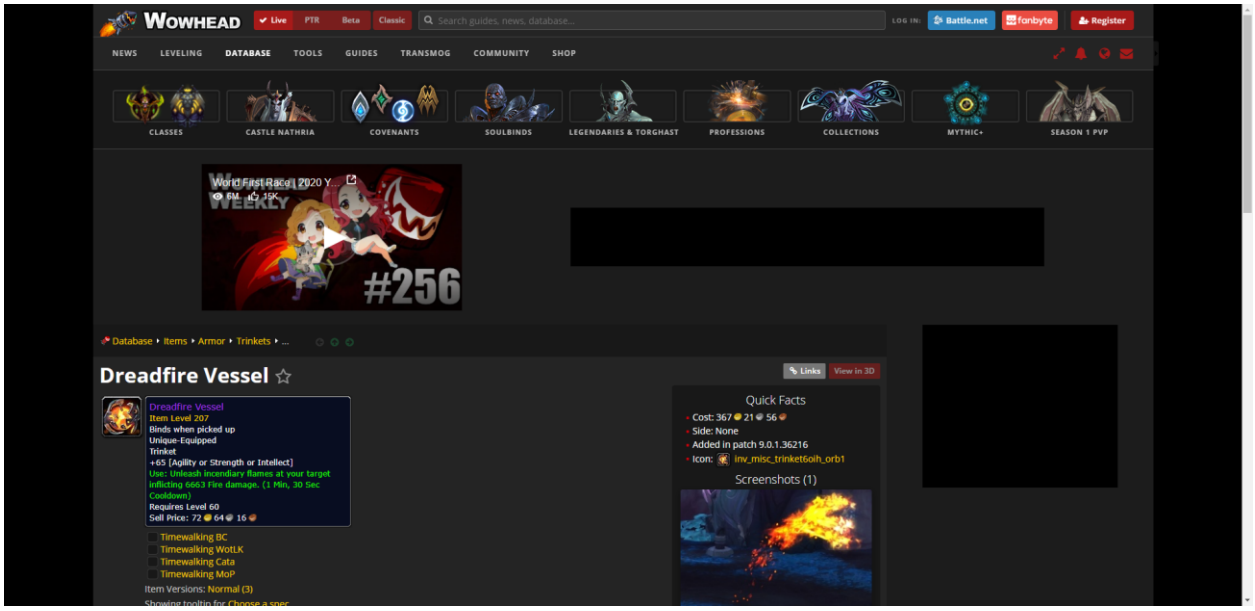
```
        driver.get(ConfigPropertiesReader.getProperty("wowhead.items"));
        driver.findElement(By.name("min-level")).sendKeys("200");
        driver.findElement(By.name("max-level")).sendKeys("210");
        Select selectSide = new Select(driver.findElement(By.name("side")));
        selectSide.selectByVisibleText("Horde");
        Select selectClass = new Select(driver.findElement(By.name("class")));
        selectClass.selectByVisibleText("Warrior");
        driver.findElement(By.className("filter-row")).submit();
        driver.findElements(By.className("listview-row")).get(0).click();
        String name = driver.findElement(By.className("heading-size-1")).getText();
        try {
            File scrFile =
((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
            FileUtils.copyFile(scrFile, new
File(".\\src\\screenshots\\FilteredItem.png"));
        } catch (IOException e) {
            System.out.println("Failed to capture screenshot: " + e.getMessage());
        }
        assertEquals("Dreadfire Vessel",name);
    }

    @AfterClass
    public static void TearDown(){
        driver.quit();
        System.out.println("Testing finished.");
    }
}
```

Скриншоты




WOWHEAD
NEWS LEVELING DATABASE TOOLS GUIDES TRANSMOG COMMUNITY SHOP
CLASSES CASTLE NATHRIA COVENANTS SOULBINDS LEGENDARIES & TORGHAST PROFESSIONS COLLECTIONS MYTHIC+ SEASON 1 PVP
#256
Guides ▸ Raids
Castle Nathria Raid Guides for Shadowlands - Boss Strategies, Rewards, Unlock Schedule
Quick Facts
By Neryssa
AUTHOR: NERYSSA
PATCH: 9.0.2
RATING: 4.4/5 (23 VOTES)


WOWHEAD
NEWS LEVELING DATABASE TOOLS GUIDES TRANSMOG COMMUNITY SHOP
CLASSES CASTLE NATHRIA COVENANTS SOULBINDS LEGENDARIES & TORGHAST PROFESSIONS COLLECTIONS MYTHIC+ SEASON 1 PVP
#256
Database ▸ Items ▸ Armor ▸ Trinkets
Inscrutable Quantum Device
Inscrutable Quantum Device
Item Level 158
Binds when picked up
Unique-Equipped
Trinket
+41 [Agility or Strength or Intellect]
Use: ??? (3 Min Cooldown)
Requires Level 60
Sell Price: 54 7 65
Dropped by: Dealer Xy'exa
Drop Chance: 25.48%
Timewalking BC
Timewalking WotLK
Timewalking Cata
Timewalking MoP
Item Versions: Normal (3)
Quick Facts
Cost: 375 16 73
Side: None
Cannot be disenchanted
Added in patch 9.0.1.36216
Icon: inv_trinket_80_titan02a
Screenshots (1)


WOWHEAD
NEWS LEVELING DATABASE TOOLS GUIDES TRANSMOG COMMUNITY SHOP
CLASSES CASTLE NATHRIA COVENANTS SOULBINDS LEGENDARIES & TORGHAST PROFESSIONS COLLECTIONS MYTHIC+ SEASON 1 PVP
#256
Database ▸ Items ▸ Armor ▸ Trinkets
Dreadfire Vessel
Dreadfire Vessel
Item Level 207
Binds when picked up
Unique-Equipped
Trinket
+65 [Agility or Strength or Intellect]
Use: Unleash incendiary flames at your target inflicting 6663 Fire damage. (1 Min, 30 Sec Cooldown)
Requires Level 60
Sell Price: 72 64 16
Timewalking BC
Timewalking WotLK
Timewalking Cata
Timewalking MoP
Item Versions: Normal (3)
Quick Facts
Cost: 367 21 56
Side: None
Added in patch 9.0.1.36216
Icon: inv_misc_trinket6oih_orb1
Screenshots (1)