ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук

Департамент программной инженерии

|  |  |
| --- | --- |
| СОГЛАСОВАНО  Научный руководитель  старший преподаватель  департамента больших данных и информационного поиска  факультета компьютерных наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Е.М. Гринкруг  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. | УТВЕРЖДЕНО  Академический руководитель  образовательной программы  «Программная инженерия»  профессор департамента программной инженерии, канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. |

|  |  |
| --- | --- |
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. № подл |  |

Сервер для мокирования асинхронных протоколов

Программа и методика испытаний

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.03.07-01 51 01-ЛУ

|  |  |
| --- | --- |
|  | Исполнители  студент группы БПИ152  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** /П.И. Данилин/  **«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  2019 г. |

УТВЕРЖДЕН

RU.17701729. 03.07-01 51 01-1-ЛУ

Сервер для мокирования асинхронных протоколов

|  |  |
| --- | --- |
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. № подл |  |

Программа и методика испытаний

RU.17701729.03.07-01 51 01-ЛУ

Листов #

# Аннотация

В данном программном документе приведена программа и методика испытаний для программного обеспечения «Сервер для мокирования асинхронных протоколов». В разделе «Объект испытаний» указаны наименование программы, а также краткая характеристика области применения. Далее, в разделе «Цель испытаний» находится цель проведения испытаний. Раздел «Требования к программе» содержит требования, подлежащие проверке в ходе испытаний. В разделе «Требования к программной документации» указан состав программной документации. В разделе «Средства и порядок испытаний» указаны технические и программные средства, используемые в ходе испытаний, а также порядок проведения испытаний. Раздел «Методы испытаний» содержит описание используемых методов испытаний, описание проверок с указанием результатов. В последнем разделе «Приложение. Перечень материалов, использованных в ходе испытаний» содержатся публикации и другие материалы, использованные в ходе испытаний.

Оформление программного документа «Программа и методика испытаний» произведено по требованиям ГОСТ 19.301-79» Руководство оператора. Требования к содержанию и оформлению» [7].

**Содержание**

[Аннотация 3](#_Toc9410467)

[1. Объект испытаний 6](#_Toc9410468)

[1.1. Наименование 6](#_Toc9410469)

[1.2. Краткая характеристика области применения программного продукта 6](#_Toc9410470)

[2. Цель испытаний 7](#_Toc9410471)

[2.1. Цель проведения испытаний 7](#_Toc9410472)

[3. Требования к программе 8](#_Toc9410473)

[3.1. Требования к функциональным характеристикам 8](#_Toc9410474)

[3.2. Требования к надежности программы 8](#_Toc9410475)

[3.3. Требования квалификация и уровня подготовки пользователя 8](#_Toc9410476)

[3.4. Требования к составу и параметрам технических средств 8](#_Toc9410477)

[3.5. Требования к информационной и программной совместимости 9](#_Toc9410478)

[4. Требования к программной документации 10](#_Toc9410479)

[4.1. Состав программной документации 10](#_Toc9410480)

[5. Средства и порядок испытаний 11](#_Toc9410481)

[5.1. Технические средства, используемые во время испытаний 11](#_Toc9410482)

[5.2. Программные средства, используемые во время испытаний 11](#_Toc9410483)

[5.3. Порядок проведения испытаний 11](#_Toc9410484)

[6. Методы испытаний 12](#_Toc9410485)

[6.1. Проверка функциональных требований 12](#_Toc9410486)

[6.1.1. Конфигурация поведения WebSocket сервера списком сообщений 12](#_Toc9410487)

[6.1.2. Конфигурация поведения WebSocket сервера реакцией списком сообщений на сообщение от клиента 13](#_Toc9410488)

[6.1.3. Конфигурация поведения WebSocket сервера комбинацией из п. 1 и 2 14](#_Toc9410489)

[6.1.4. Конфигурация поведения WebSocket с помощью конфигурационного файла при старте сервера 15](#_Toc9410490)

[6.1.5. Конфигурация поведения WebSocket с помощью REST API 16](#_Toc9410491)

[6.1.6. Добавление задержки перед отправкой события в WebSocket 16](#_Toc9410492)

[6.1.7. Конфигурация поведения Server-Sent Events сервера списком сообщений 17](#_Toc9410493)

[6.1.8. Конфигурация поведения Servers-Sent Events с помощью конфигурационного файла при старте сервера 17](#_Toc9410494)

[6.1.9. Конфигурация поведения Servers-Sent Events с помощью REST API 17](#_Toc9410495)

[6.1.10. Добавление задержки перед отправкой события в Server-Sent Events 17](#_Toc9410496)

[6.2. Проверка требований к надежности 17](#_Toc9410497)

[6.2.1. Программа должна осуществлять проверку корректности введенных пользователем данных и выводить сообщение об ошибке при ее нахождении 17](#_Toc9410498)

[6.2.2. Ошибки, возникающие в приложении 17](#_Toc9410499)

[7. Список использованной литературы 18](#_Toc9410500)

# Объект испытаний

## Наименование

Наименование программного продукта – «Сервер для мокирования асинхронных протоколов».

## Краткая характеристика области применения программного продукта

Разрабатываемый программный продукт можно применить при разработке сервера с сетевым интерфейсом, в котором используется протокол WebSocket или Server-Sent Events. А также при разработке приложения, которое использует подобный интерфейс. Пользователь сможет создать имитацию реального сервиса путем описания желаемого поведения, без его непосредственной реализации.

# Цель испытаний

## Цель проведения испытаний

Целью проведения испытаний является проверка корректности работы программы, а также проверка соответствия разработанной программы функциональным требованиям и требованиям к надежности, изложенным в документе «Техническое задание».

# Требования к программе

## Требования к функциональным характеристикам

Создаваемый программный продукт должен иметь следующую функциональность:

1. Конфигурация поведения WebSocket сервера списком сообщений
2. Конфигурация поведения WebSocket сервера реакцией списком сообщений списком сообщений на сообщение от клиента
3. Конфигурация поведения WebSocket сервера комбинацией из п. 1 и 2
4. Конфигурация поведения WebSocket с помощью конфигурационного файла при старте сервера
5. Конфигурация поведения WebSocket с помощью REST API
6. Добавление задержки перед отправкой события в WebSocket
7. Конфигурация поведения Server-Sent Events сервера списком сообщений
8. Конфигурация поведения Servers-Sent Events с помощью конфигурационного файла при старте сервера
9. Конфигурация поведения Servers-Sent Events с помощью REST API
10. Добавление задержки перед отправкой события в Server-Sent Events

## Требования к надежности программы

К программе предъявляются следующие требования надежности:

Программа должна осуществлять проверку корректности введенных пользователем данных и выводить сообщение об ошибке при ее нахождении.

Программа должна корректно завершаться при возникновении ошибок.

## Требования квалификация и уровня подготовки пользователя

Пользователь должен иметь базовые навыки работы в текстовых редакторах, а также уметь читать и редактировать JSON файлы.

Обязательно ознакомление пользователя с руководством оператора.

## Требования к составу и параметрам технических средств

Для эффективной работы с программой предъявляются следующие требования к составу и параметрам технических средств:

1. Процессор Pentium 2 266 MHz или мощнее
2. Оперативная память не менее 1 Гб
3. Свободное дисковое пространство не меньше 1 Гб
4. Исправная сетевая карта
5. Монитор
6. Клавиатура
7. Мышь

## Требования к информационной и программной совместимости

Для корректной работы программного продукта требуется:

1. Установленный JDK версии 1.8 или выше
2. Установленный SBT версии 1.2.8 или совместимый
3. Установленный Docker версии 2.0.0.3 или совместимый

# Требования к программной документации

## Состав программной документации

На испытания должна быть представлена документация к программе в следующем составе:

1. «Сервер для мокирования асинхронных протоколов». Техническое задание. ГОСТ 19.71-208
2. «Сервер для мокирования асинхронных протоколов». Руководство оператора. ГОСТ 19.505-79
3. «Сервер для мокирования асинхронных протоколов». Программа и методика испытаний. ГОСТ 19.301-79
4. «Сервер для мокирования асинхронных протоколов». Текст программы ГОСТ 19.401-78

# Средства и порядок испытаний

## Технические средства, используемые во время испытаний

Для надежной и бесперебойной работы программы требуется следующий состав технических средств:

1. Процессор Intel Core i5 2500
2. Оперативная память 4 Гб
3. Свободное дисковое пространство 128 Гб
4. Исправная сетевая карта
5. Монитор
6. Клавиатура
7. Мышь

## Программные средства, используемые во время испытаний

Для работы программы необходим следующий состав программных средств:

1. JDK версии 1.8
2. SBT версии 1.2.8
3. Docker версии 2.0.0.3

## Порядок проведения испытаний

Испытания должны проводиться в следующем порядке:

1. Проверка требований к программной документации
2. Проверка требований к функциональным характеристикам
3. Проверка требований к надёжности

# Методы испытаний

## Проверка функциональных требований

Чтобы запустить программу требуется ввести в консоль команду:

docker build -t async-mock . **&&** docker run -it --rm -p 2525:2525 -p 5000:5000 async-mock

После этого требуется открыть браузер, и в адресной строке ввести адрес localhost:2525/swagger

### Конфигурация поведения WebSocket сервера списком сообщений

Создадим WebSocket мок со следующей конфигурацией и отправим его на сервер (рис. 1):

{  
 **"port"**: 5000,  
 **"stubs"**: [  
 {  
 **"predicates"**: [],  
 **"responses"**: [  
 {  
 **"events"**: [  
 {  
 **"data"**: **"Hello World!"** }  
 ],  
 **"type"**: **"websocket"** }  
 ]  
 }  
 ]  
}

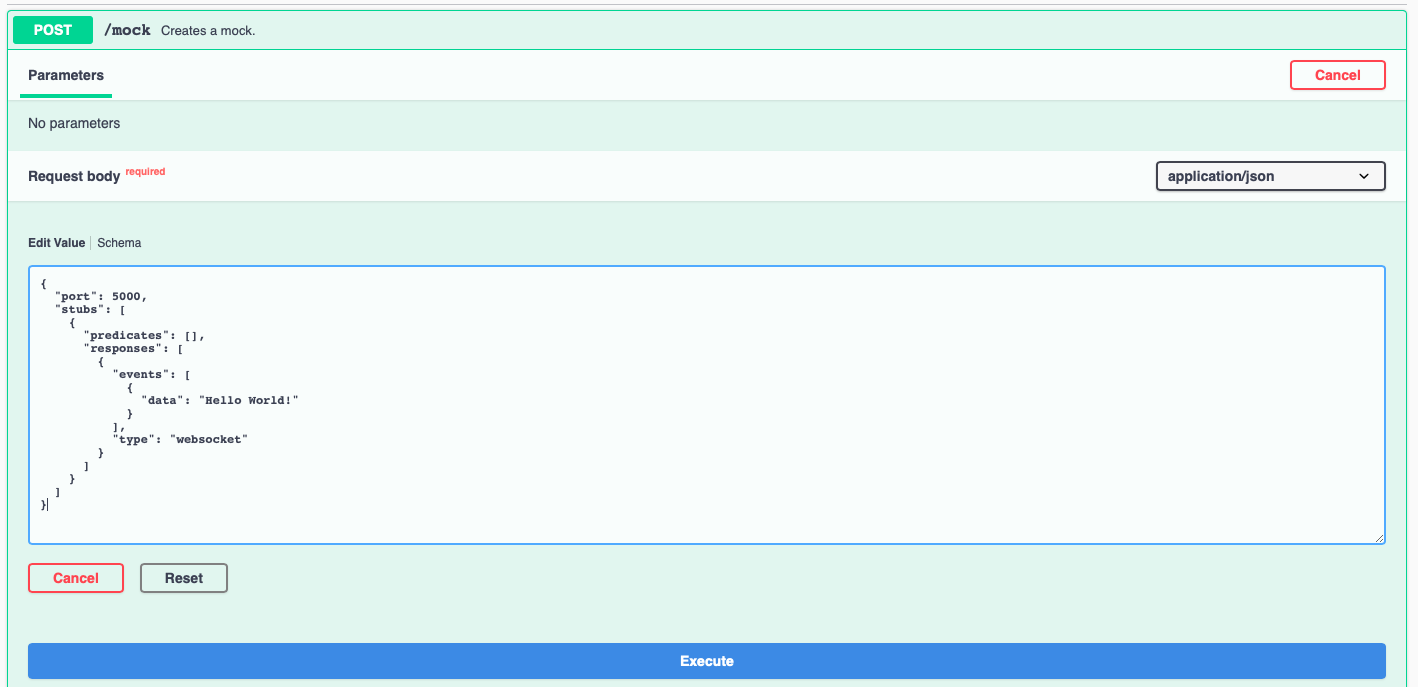


Рисунок 1. Отправка мока с последовательностью событий для WebSocket

Для проверки WebSocket будем использовать утилиту wsc [1]. Следующей командой проверим работоспособность мока:

wsc -er ws://localhost:5000

Вывод программы:

Connected to ws://localhost:5000

< "Hello World!" (5ms)

После каждой проверки будем удалять созданный мок.

### Конфигурация поведения WebSocket сервера реакцией списком сообщений на сообщение от клиента

Создадим WebSocket мок со следующей конфигурацией и отправим его на сервер (рис. 2):

{  
 **"port"**: 5000,  
 **"stubs"**: [  
 {  
 **"predicates"**: [],  
 **"responses"**: [  
 {  
 **"reactions"**: [  
 {  
 **"predicates"**: [],  
 **"reaction"**: [  
 {  
 **"data"**: **"response"** }  
 ]  
 }  
 ],  
 **"events"**: [],  
 **"type"**: **"websocket"** }  
 ]  
 }  
 ]  
}

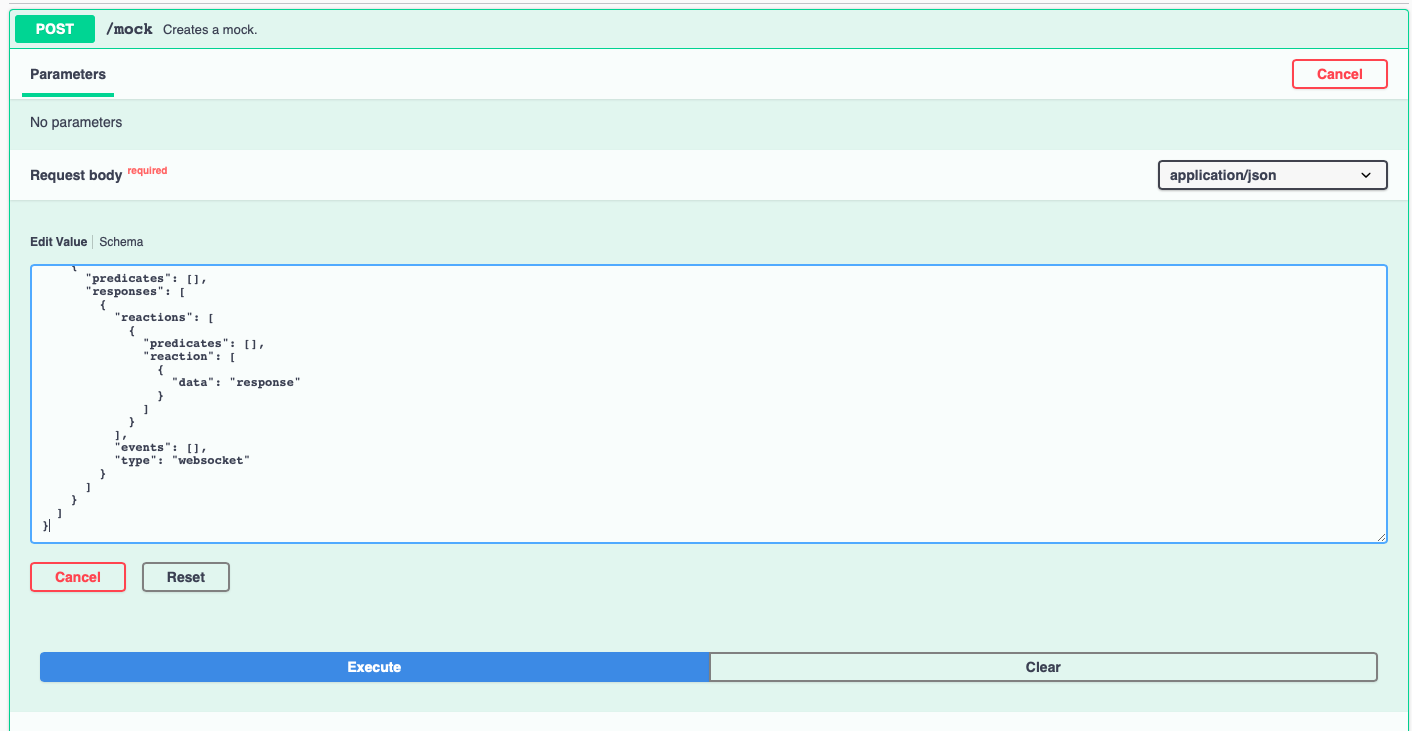


Рисунок 2. Отправка мока с реакцией на события от клиента для WebSocket

Следующей командой проверим работоспособность мока:

wsc -er ws://localhost:5000

Вывод программы:

Connected to ws://localhost:5000

> Test event

< "response" (80ms)

### Конфигурация поведения WebSocket сервера комбинацией из п. 1 и 2

Создадим WebSocket мок со следующей конфигурацией и отправим его на сервер (рис. 3):

{  
 **"port"**: 5000,  
 **"stubs"**: [  
 {  
 **"predicates"**: [],  
 **"responses"**: [  
 {  
 **"reactions"**: [  
 {  
 **"predicates"**: [],  
 **"reaction"**: [  
 {  
 **"data"**: **"response"** }  
 ]  
 }  
 ],  
 **"events"**: [  
 {  
 **"data"**: **"Hello World!"** }  
 ],  
 **"type"**: **"websocket"** }  
 ]  
 }  
 ]  
}

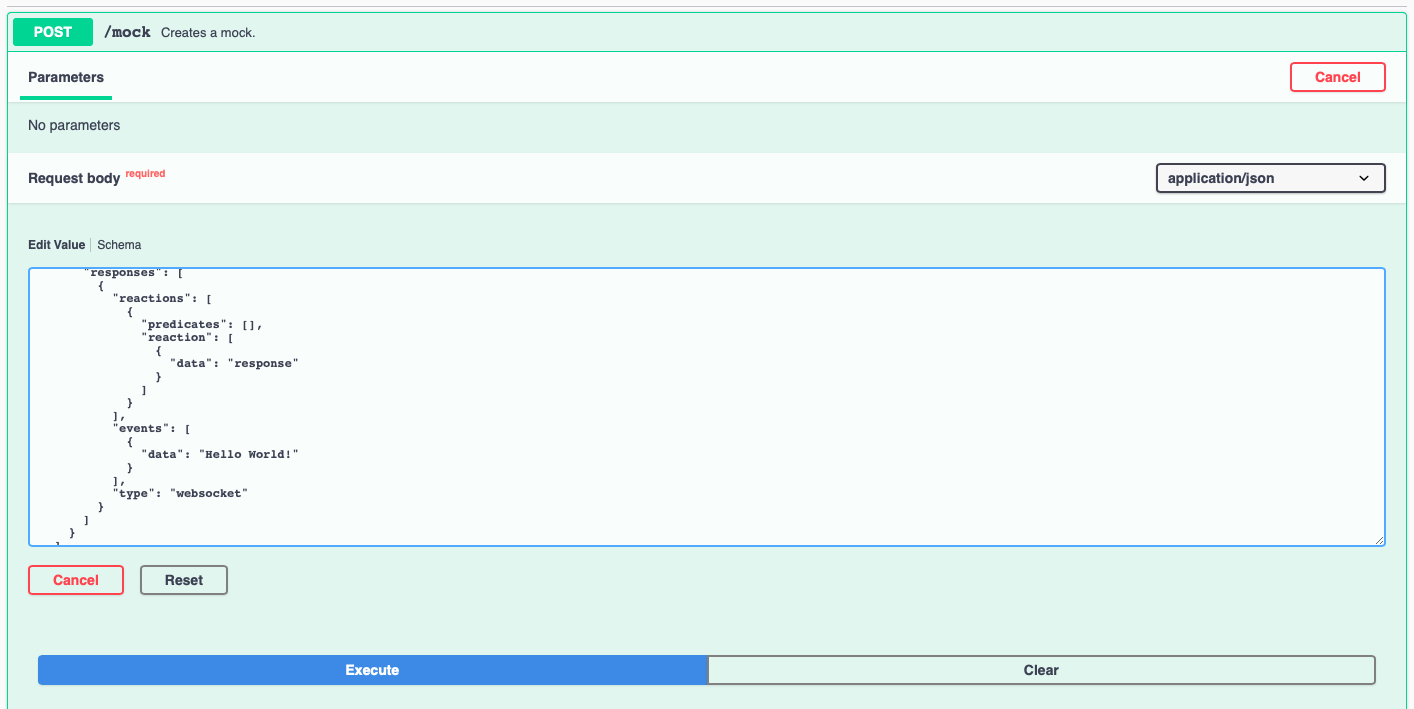


Рисунок 3. Отправка мока с последовательностью событий и реакцией на события от клиента для WebSocket

Следующей командой проверим работоспособность мока:

wsc -er ws://localhost:5000

Вывод программы:

Connected to ws://localhost:5000

< "Hello World!" (5ms)

> Test event

< "response" (28ms)

### Конфигурация поведения WebSocket с помощью конфигурационного файла при старте сервера

Создадим файл configuration.json. В него запишем конфигурацию из п. 6.1.1. Далее запустим сервер, передав аргумент -f configuration.json в параметры запуска. При запуске приложение выводит в лог:

Starting AsyncMock with args [-f, configuration.json]

Проверим работоспособность мока (вывод wsc):

Connected to ws://localhost:5000

< "Hello World!" (5ms)

### Конфигурация поведения WebSocket с помощью REST API

Данное требование также проверяется в проверках 6.1.1 – 6.1.3.

### Добавление задержки перед отправкой события в WebSocket

Создадим WebSocket мок со следующей конфигурацией и отправим его на сервер (рис. 4):

{  
 **"port"**: 5000,  
 **"stubs"**: [  
 {  
 **"predicates"**: [],  
 **"responses"**: [  
 {  
 **"reactions"**: [  
 {  
 **"predicates"**: [],  
 **"reaction"**: [  
 {  
 **"data"**: **"response"**,  
 **"delay"**: **"1 second"** }  
 ]  
 }  
 ],  
 **"events"**: [  
 {  
 **"data"**: **"Hello World!"**,  
 **"delay"**: **"2 seconds"** }  
 ],  
 **"type"**: **"websocket"** }  
 ]  
 }  
 ]  
}

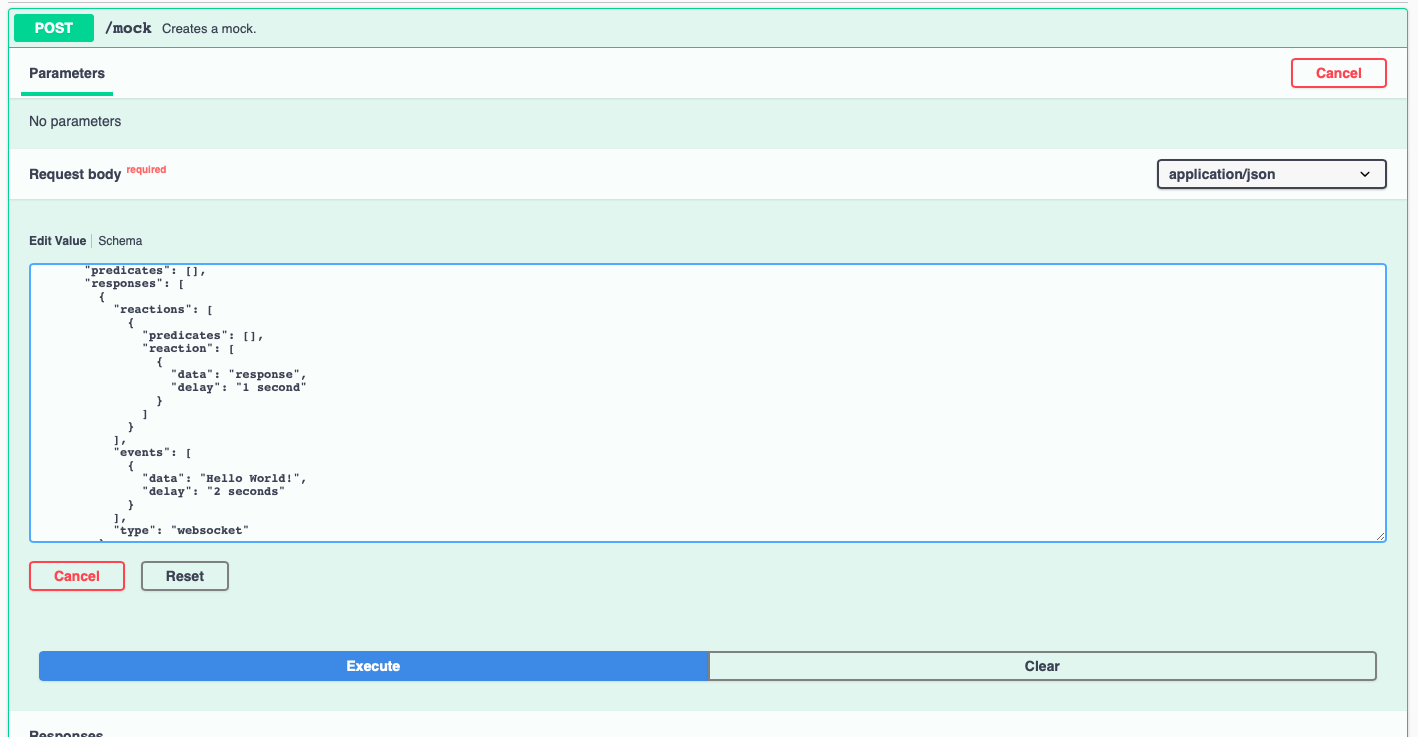


Рисунок 4. Отправка мока с задержкой для WebSocket

Для проверки WebSocket будем использовать утилиту wsc [1]. Следующей командой проверим работоспособность мока:

wsc -er ws://localhost:5000

Вывод программы:

Connected to ws://localhost:5000

< "Hello World!" (2016ms)

> Test event

< "response" (1029ms)

### Конфигурация поведения Server-Sent Events сервера списком сообщений

### Конфигурация поведения Servers-Sent Events с помощью конфигурационного файла при старте сервера

### Конфигурация поведения Servers-Sent Events с помощью REST API

### Добавление задержки перед отправкой события в Server-Sent Events

## Проверка требований к надежности

### Программа должна осуществлять проверку корректности введенных пользователем данных и выводить сообщение об ошибке при ее нахождении

### Ошибки, возникающие в приложении

# Список использованной литературы

1. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. ГОСТ Р 7.02-2006 Консервация документов на компакт-дисках. Общие требования. – М.: ИПК Издательство стандартов, 2006
11. ГОСТ 18300-87 Спирт этиловый ректификованный технический. Технические условия. – М.:ИПК Издательство стандартов, 1997
12. ГОСТ 9805-84. Спирт изопропиловый. Технические условия. – М.: ИПК Издательство стандартов, 1984.
13. ГОСТ 19.602-78 Правила дублирования, учета и хранения программных документов, выполненных печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001