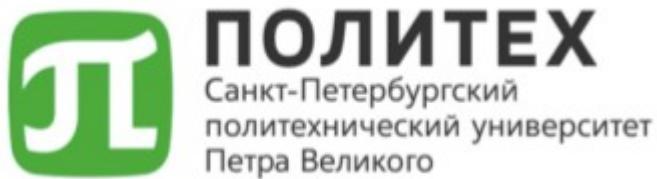


Федеральное государственное автономное образовательное учреждение высшего образования
«Санкт-Петербургский политехнический университет Петра Великого»
Институт компьютерных наук и кибербезопасности
Программная инженерия



Отчет по курсовой работе

по дисциплине «Конструирование программного обеспечения»

на тему «**Инструмент коллаборативного автоматизированного перевода
(CAT-программа)**»

Руководитель

Юркин В. А.

Санкт-Петербург
2026

Оглавление

Оглавление.....	2
Woofie.....	3
Участники проекта.....	3
Технологический стек.....	3
Определение проблемы.....	3
Основные требования.....	4
Разработка архитектуры и детальное проектирование.....	4
Характер нагрузки.....	4
API.....	5
1. Работа с текстом.....	5
GET /translations.....	5
POST /translations-fetch-updates.....	5
POST /translations.....	6
GET /translations/{id}.....	7
PUT /translations/{id}.....	7
DELETE /translations/{id}.....	8
2. Работа с глоссарием.....	8
GET /glossary.....	8
POST /glossary-fetch-updates.....	9
POST /glossary.....	9
GET /glossary/{id}.....	10
PUT /glossary/{id}.....	10
DELETE /glossary/{id}.....	10
3. Машинный перевод.....	11
GET /translation-services.....	11
POST /translate/{id}.....	11
Демонстрация работы.....	12
Тестирование.....	12
Тестирование с нагрузкой.....	18
Сборка и запуск.....	20

Woofie

Инструмент колаборативного автоматизированного перевода (CAT-программа)

GitHub Репозиторий Woofie: <https://github.com/odintsovks/Woofie>

GitHub Репозиторий Woofie-Server: <https://github.com/odintsovks/Woofie-Server>

GitHub Репозиторий Woofie-Client: <https://github.com/odintsovks/Woofie-Client>

Участники проекта

Одинцов Кирилл Сергеевич (гр. 5130904/30106)

Дмитриев Арсений Эдуардович (гр. 5130904/30106)

Засульский Егор Андреевич (гр. 5130904/30106)

Маланьин Никита Александрович (гр. 5130904/30106)

Проселков Станислав Павлович (гр. 5130904/30106)

Горбунова Алена Евгеньевна (гр. 5130904/30106)

Технологический стек

- Серверная часть: Java Spring Framework
- Базы данных: PostgreSQL
- Внешние зависимости: интеграция с сервисами машинного перевода (к примеру «Яндекс.Переводчик» с использованием API-ключа)
- UI: Qt 6

Определение проблемы

Отсутствие открытых CAT-программ позволяющих организовывать проекты в виде дерева. Вместо этого существующие инструменты ограничиваются табличным отображением единиц перевода, что осложняет процесс, когда последовательность текста зависит от конкретных условий (к примеру «деревья диалогов» в видеоиграх).

Основные требования

- Возможность использования древовидной структуры для отображения единиц перевода.

Когда на практике переводимый текст зависит от строго заданных условий, я хочу иметь возможность отражать это структурой единиц перевода в САТ-программе, чтобы не путаться в исключительно последовательном табличном отображении

- Возможность делать запросы в сервисы машинного перевода и проверки правописания.

Когда возникает необходимость сверить смысл или правописание, я хочу использовать быстродоступные сервисы машинного перевода и правописания, чтобы не выбиваться из процесса перевода.

- Возможность редактировать единицы перевода в режиме реального времени.

Когда другой пользователь тем или иным образом изменил единицу перевода, я хочу увидеть это изменение без нужды ручного обновления, чтобы всегда видеть актуальное состояние проекта.

Ожидается, что данным набором инструментов (клиентом и сервером) будут пользоваться команды переводчиков текста, которые обычно являются небольшими, но программы строятся с рассчётом на максимум 10к пользователей в сутки. Как правило проекты завершаются меньше, чем за год, период хранения информации соответствующий.

Разработка архитектуры и детальное проектирование

Характер нагрузки

- R/W нагрузка - ожидается более высокая интенсивность чтения. При работе с текстом программа-клиент будет активно производить операции вставки в базу данных, но при этом текст должен автоматически синхронизироваться между всеми клиентами, таким образом чтение каждого нового изменения в базу данных растёт с количеством пользователей.
- Объём трафика - подавляющая большую часть трафика будут занимать текстовые данные единиц перевода, оценить объём можно следующим образом:
 - Одну единицу перевода можно оценить в 1 КиБ (достаточно примерно на 1024 / 2 байта на символ / 4 букв на слово = 128 слов в кодировке UTF-8), один пользователь за рабочую неделю сделает примерно 500 изменений в текст (40 часов * 60 / 5 минут на изменение = 480 изменений) изменений в текст, соответственно количество данных на запись оценивается в 1 КиБ * 500 / 7 = 72 КиБ на пользователя в день. При этом количество данных на чтение дополнительно растёт с количеством пользователей, т.к. каждый пользователь должен иметь доступ к истории изменений.
 - Запись:
 - Типовой случай: 72 КиБ/день * 100 пользователей = 7200 КиБ/день = 7 МиБ/день
 - Худший случай: 72 КиБ/день * 10000 пользователей = 720000 КиБ/день = 700 МиБ/день
 - Чтение:
 - Типовой случай: 7 МиБ/день * 100 пользователей = 700 МиБ/день
 - Худший случай: 7 МиБ/день * 10000 пользователей = 70000 МиБ/день = 70 ГиБ/день

- На практике количество данных на запись/чтение будет ограничено сверху количеством единиц перевода.
- Объёмы дисковой системы - как в случае с объёмом трафика, большую часть дискового пространства будут занимать текстовые данные; в зависимости от размера проекта количество единиц перевода может превышать десятки, если не сотни тысяч. Как правило одна единица требует не больше двух-трёх итераций (черновой вариант + вычитка), соответственно:
 - Типовой случай: 1 КиБ/ед.п. * 10000 ед.п. * 3 итерации = 30000 КиБ = 29 МиБ
 - Худший случай: 1 КиБ/ед.п. * 100000 ед.п. * 3 итерации = 300000 КиБ = 293 МиБ

API

1. Работа с текстом

GET /translations

Описание: запрос всех единиц перевода (пар "оригинал-перевод")

- Статус: 200 OK
 - Тело:

```
{
  "timestamp": 0,
  "translations": [
    {
      "id": "",
      "sourceText": "",
      "targetText": "",
      "connections": [
        {
          "id": "",
          "description": ""
        },
        // ...
      ],
      // ...
    }
  ]
}
```

Требования по максимальному времени отклика: 5 секунд (запрос производится при запуске клиента, ожидаем что формирование запроса может быть медленным, а время отклика некритично на этапе инициализации программы).

POST /translations-fetch-updates

Описание: запрос обновлённых единиц перевода с определённой отметкой времени

- Тело запроса:


```
{
  "timestamp": 0
}
```
- Статус: 200 OK

- Тело:


```
[
    {
      "id": "",
      "sourceText": "",
      "targetText": "",
      "connections": [
        {
          "id": "",
          "description": ""
        },
        // ...
      ],
      // ...
    }
  ]
```
- Статус: *400 Bad Request*
 - Неправильно сформирован запрос; отсутствует поле *timestamp*

Требования по максимальному времени отклика: 1 секунда (ожидается, что программа будет запрашивать свежие обновления текстовых данных раз в 1-2 секунды)

POST /translations

Описание: добавить единицу перевода

- Тело запроса:


```
{
  "sourceText": "",
  "targetText": "",
  "connections": [
    {
      "id": "",
      "description": ""
    },
    // ...
  ]
}
```
- Статус: *201 Created*
 - Тело:

```
{
  "id": "",
  "sourceText": "",
  "targetText": "",
  "connections": [
    {
      "id": "",
      "description": ""
    },
    // ...
  ]
}
```

- Статус: *400 Bad Request*
 - Неправильно сформирован запрос; отсутствует одно из полей

GET /translations/{id}

Описание: запросить единицу перевода по заданному индексу

Статус: *200 OK*

- Тело:

```
{
  "sourceText": "",
  "targetText": "",
  "connections": [
    {
      "id": "",
      "description": ""
    },
    // ...
  ]
}
```

- Статус: *404 Not Found*

• Единицы перевода не существует по заданному индексу

PUT /translations/{id}

Описание: установить значения единицы перевода по заданному индексу (добавить если не существует)

- Тело запроса:

```
{
  "sourceText": "",
  "targetText": "",
  "connections": [
    {
      "id": ""
    }
  ]
}
```

```

    "description": "",

},
// ...
]

}

• Статус: 200 OK и 201 Created

◦ Тело
{
  "id": "",
  "sourceText": "",
  "targetText": "",
  "connections": [
    {
      "id": "",
      "description": ""
    },
    // ...
  ]
}

```

- Статус: *400 Bad Request*
 - Неправильно сформирован запрос; отсутствует одно из полей.

DELETE /translations/{id}

Описание: удалить единицу перевода по заданному индексу

- Статус: *204 No Content*
- Статус: *404 Not Found*
 - Единицы перевода не существовало по заданному индексу перед запросом на удаление

2. Работа с глоссарием

GET /glossary

Описание: запрос списка записей глоссария

- Статус: *200 OK*
 - Тело:


```
{
  "timestamp": 0,
  "entries": [
    {
      "id": "",
      "targetTerm": "",
      "sourceTerm": "",
      "definition": ""
    },
    // ...
  ]
}
```

```
        // ...
    ]
}
```

Требования по максимальному времени отклика: 5 секунда (аналогично с GET /translations)

POST /glossary-fetch-updates

Описание: запрос обновлённых записей глоссария с определённой отметки времени

- Тело запроса:

```
{
  "timestamp": 0
}
```

- Статус: *200 OK*

Тело:

```
[
  {
    "id": "",
    "targetTerm": "",
    "sourceTerm": "",
    "definition": ""
  },
  // ...
]
```

- Статус: *400 Bad Request*

- Неправильно сформирован запрос; отсутствует поле *timestamp*

Требования по максимальному времени отклика: 5 секунд (аналогично с GET /translations-fetch-updates, но ожидаем, что обновления глоссария запрашиваются реже).

POST /glossary

Описание: добавить запись в глоссарий.

- Тело запроса:

```
{
  "targetTerm": "",
  "sourceTerm": "",
  "definition": ""
}
```

- Статус: *201 Created*

- Тело:

```
{
  "id": "",
```

```
        "targetTerm": "",  
        "sourceTerm": "",  
        "definition": ""  
    }  
}
```

- Статус: *400 Bad Request*

- Неправильно сформирован запрос; отсутствует одно из полей.

GET /glossary/{id}

Описание: запросить запись из глоссария по индексу

- Статус: *200 OK*

- Тело:

```
{  
    "targetTerm": "",  
    "sourceTerm": "",  
    "definition": ""  
}
```

- Статус: *404 Not Found*

- Записи глоссария не существует по заданному индексу

PUT /glossary/{id}

Описание: изменить запись глоссария по индексу (добавить если не существует)

- Тело запроса:

```
{  
    "targetTerm": "",  
    "sourceTerm": "",  
    "definition": ""  
}
```

- Статус: *200 OK* и *201 Created*

- Тело

```
{  
    "id": "",  
    "targetTerm": "",  
    "sourceTerm": "",  
    "definition": ""  
}
```

- Статус: *400 Bad Request*

- Неправильно сформирован запрос; отсутствует одно из полей.

DELETE /glossary/{id}

Описание:

- Статус: *204 No Content*
- Статус: *404 Not Found*
 - Записи глоссария не существовало по заданному индексу перед запросом на удаление

3. Машинный перевод

GET /translation-services

Описание: запросить информацию о всех доступных сервисах машинного перевода

- Статус: *200 OK*
 - Тело:


```
[
            {
              "id": "",
              "name": ""
            },
            //...
          ]
```

POST /translate/{id}

Описание: запросить перевод строки у сервиса машинного перевода по заданному индексу

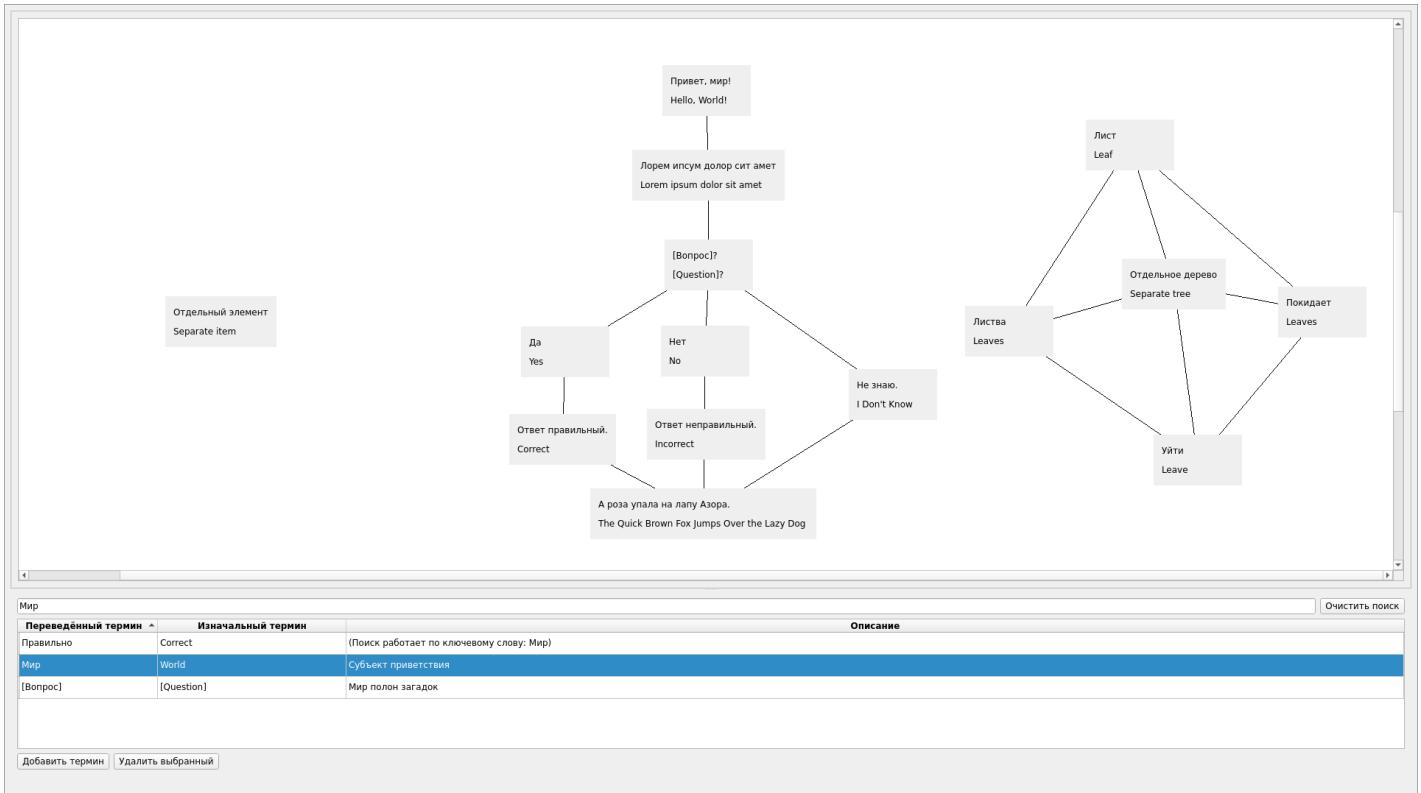
- Тело запроса:


```
{
  "text": ""
}
```
- Статус: *200 OK*
 - Тело:


```
{
  "text": ""
}
```
- Статус: *503 Service Unavailable*
 - Сервис перевода не доступен; сервер не получил ответ

Требования по максимальному времени отклика: N/a (полностью зависит от доступности и времени отклика стороннего сервиса)

Демонстрация работы



Тестирование

Тестовые запросы http к серверу:

1. GET /api/translations (Запрос всех единиц)

Ожидаемый статус: 200 OK

Ожидаемое тело: { "timestamp": ..., "translations": [...] }

GET http://localhost:8080/api/translations

2. POST /api/translations (Добавление единицы)

Ожидаемый статус: 201 Created

POST http://localhost:8080/api/translations

Content-Type: application/json

{

 "sourceText": "Hello world",

```
"targetText": "Привет мир",
"connections": [
  { "description": "First connection" }
]
}
```

3. POST /api/translations (Ошибка 400 - отсутствует поле)

Ожидаемый статус: *400 Bad Request*

POST http://localhost:8080/api/translations

Content-Type: application/json

```
{
  "targetText": "Only target"
}
```

4. GET /api/translations/{id} (Запрос по ID)

Ожидаемый статус: *200 OK*

GET http://localhost:8080/api/translations/2

5. DELETE /api/translations/{id} (Удаление)

Ожидаемый статус: *204 No Content*

DELETE http://localhost:8080/api/translations/1

6. DELETE /api/translations/{id} (Ошибка 404 - уже удалено)

Ожидаемый статус: *404 Not Found*

DELETE http://localhost:8080/api/translations/1

```

Project Alt+1
resources application.properties
Run GlossaryServiceApplication Tests in 'woofie-server.app.test' Tests in 'com.woofie.glossary'

Test Results 726ms
3 tests passed 3 tests total, 726 ms

Calculating task graph as configuration cache cannot be reused because system property 'idea.io.use.nio2' has changed.
> Task :app:processTestResources NO-SOURCE
> Task :app:processResources UP-TO-DATE
> Task :app:compileTestJava UP-TO-DATE
> Task :app:testClasses UP-TO-DATE
> Task :app:processTestClasses
> Task :app:testClasses
16:36:15.890 [Test worker] INFO org.springframework.test.context.support.AnnotationConfigContextLoaderUtils -- Could not detect default configuration classes for test class [com.woofie.glossary.GlossaryIntegrationTest]
16:36:15.978 [Test worker] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper -- Found @SpringBootConfiguration com.woofie.glossary.GlossaryServiceApplication for test class com.woofie.glossary.GlossaryIntegrationTest
16:36:15.978 [Test worker] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper -- Found @SpringBootConfiguration com.woofie.glossary.GlossaryServiceApplication for test class com.woofie.glossary.GlossaryIntegrationTest

:: Spring Boot :: (v3.4.1)

2026-01-09T16:36:16.347+03:00 INFO 7528 --- [ Test worker] c.w.glossary.GlossaryIntegrationTest : Starting GlossaryIntegrationTest using Java 21.0.9 with PID 7528 (started by PC in C:\Users\PC\IdeaProjects\Woofie-Server\app)
2026-01-09T16:36:16.348+03:00 INFO 7528 --- [ Test worker] c.w.glossary.GlossaryIntegrationTest : No active profile set, falling back to 1 default profile: "default"
2026-01-09T16:36:16.862+03:00 INFO 7528 --- [ Test worker] i.d.s.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repository configuration. Located at [file:/C:/Users/PC/IdeaProjects/Woofie-Server/app/src/main/java/com/woofie/glossary/jpa/repository/config/RepositoryConfigurationDelegate.java]
2026-01-09T16:36:16.863+03:00 INFO 7528 --- [ Test worker] i.d.s.c.RepositoryConfigurationDelegate : Repository configuration annotated with @EnableJpaRepositories found 2 JPA repository interfaces.
2026-01-09T16:36:17.268+03:00 INFO 7528 --- [ Test worker] o.h.b.Hibernate.jpa.ConnectionUtil$LogHelper : MM00002046: Processing PersistenceUnitInfo [name: default]
2026-01-09T16:36:17.325+03:00 INFO 7528 --- [ Test worker] org.hibernate.Version : MM00000426: Hibernate ORM core version 6.6.4.Final
2026-01-09T16:36:17.358+03:00 INFO 7528 --- [ Test worker] o.h.c.internal.RegionStatisticsInitiator : MM00000266: Second-level cache disabled
2026-01-09T16:36:17.418+03:00 INFO 7528 --- [ Test worker] o.s.o.p.SpringBatchItemInfo : No loadTimeWeaver setup: ignoring JPA class transformer
2026-01-09T16:36:17.443+03:00 INFO 7528 --- [ Test worker] o.s.o.p.SpringBatchItemProcessor : HikariPool-1 - Added connection org.postgresql.jdbc.PgConnection@7e0d9f1
2026-01-09T16:36:17.915+03:00 INFO 7528 --- [ Test worker] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2026-01-09T16:36:17.967+03:00 INFO 7528 --- [ Test worker] org.hibernate.orm.connections.Pool : HHH01001005: Database info:
Database JDBC URL [Connecting through datasource 'HikariDataSource (HikariPool-1)']
Database driver: undefined/unknown
Database version: 17.7
Autocommit: true
Auto-commit: true/unknown
Isolation level: undefined/unknown
Minimum pool size: undefined/unknown
Maximum pool size: undefined/unknown
2026-01-09T16:36:18.721+03:00 INFO 7528 --- [ Test worker] o.h.e.j.p.l.JtaPlatformIntegration : MM0000489: No JTA platform available (set 'hibernate.transaction.jta.platform' to enable JTA platform integration)
2026-01-09T16:36:18.722+03:00 INFO 7528 --- [ Test worker] o.h.e.j.p.l.JtaPlatformIntegration : MM0000489: No JTA EntityManagerFactory for persistence unit 'default'
2026-01-09T16:36:18.723+03:00 INFO 7528 --- [ Test worker] o.h.e.j.p.l.JtaPlatformIntegration : Spring 2.0.0 open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly config
2026-01-09T16:36:18.724+03:00 INFO 7528 --- [ Test worker] o.s.b.a.w.SpringBatchTaskServiceContext : Initializing Spring TestDispatcherServlet
2026-01-09T16:36:19.457+03:00 INFO 7528 --- [ Test worker] o.s.t.web.servlet.TestDispatcherServlet : Initializing Servlet ''
2026-01-09T16:36:19.459+03:00 INFO 7528 --- [ Test worker] o.s.t.web.servlet.TestDispatcherServlet : Completed initialization in 2 ms
2026-01-09T16:36:19.480+03:00 INFO 7528 --- [ Test worker] c.w.glossary.GlossaryIntegrationTest : Started GlossaryIntegrationTest in 3.543 seconds (process running For 4.295)

```

```

2026-01-09T16:36:20.155+03:00 INFO 7528 --- [ Test worker] t.c.s.AnnotationConfigContextLoaderUtils : Could not detect default configuration classes for test class [com.woofie.glossary.TranslationIntegrationTest]
2026-01-09T16:36:20.157+03:00 INFO 7528 --- [ Test worker] b.t.c.SpringBootTestTestContextBootstrapper : Found @SpringBootConfiguration com.woofie.glossary.GlossaryServiceApplication for test class com.woofie.glossary.TranslationIntegrationTest
Hibernate: insert into translations (source_text,target_text,updated_at) values (?,?)
Hibernate: insert into translation_connections (description,translation_id) values (?)
Hibernate: insert into translation_connections (description,translation_id) values (?)
Hibernate: select t1_0.id,t1_0.source_text,t1_0.target_text,t1_0.updated_at from translations t1_0
Hibernate: select c1_0.translation_id,c1_0.id,c1_0.description from translation_connections c1_0 where c1_0.translation_id=?
Hibernate: select c1_0.translation_id,c1_0.id,c1_0.description from translation_connections c1_0 where c1_0.translation_id=?
Hibernate: select c1_0.translation_id,c1_0.id,c1_0.description from translation_connections c1_0 where c1_0.translation_id=?
Hibernate: select t1_0.id,t1_0.source_text,t1_0.target_text,t1_0.updated_at from translations t1_0 where t1_0.id=?
2026-01-09T16:36:20.235+03:00 INFO 7528 --- [ionShutdownHook] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit 'default'
2026-01-09T16:36:20.237+03:00 INFO 7528 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated...
2026-01-09T16:36:20.240+03:00 INFO 7528 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown completed.
> Task :app:test
BUILD SUCCESSFUL in 8s
4 actionable tasks: 2 executed, 2 up-to-date

```

```

Project ~
Services
HTTP Request All in test: passed: 12 of 12 tests
Docker

HTTP Requests 600ms
All in test: passed: 12 of 12 tests
HTTP Requests 600ms
# Stopped. 6 tests passed 6 / 12 tests, 600 ms

Testing started at 10:39 ...
GET http://localhost:8080/api/translations
##

HTTP/1.1 200
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Date: Fri, 09 Jan 2026 13:39:05 GMT

Response code: 200; Time: 255ms (255 ms); Content length: 471 bytes (471 B)

POST http://localhost:8080/api/translations
Content-Type: application/json

{
  "sourceText": "Hello world",
  "targetText": "Power WP",
  "connections": [
    { "description": "First connection" }
  ]
}
##

HTTP/1.1 201
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Date: Fri, 09 Jan 2026 13:39:05 GMT

```

```
HTTP Requests 600ms ✓ Stopped. 6 tests passed 6 / 12 tests, 600ms
  1. GET /api/translatic 255ms
    ✓ Request
    ✓ Response 255ms
  #2
    ✓ Request
    ✓ Response 78ms
  #3
    ✓ Request
    ✓ Response 5ms
  #4
    ✓ Request
    ✓ Response 13ms
  #5
    ✓ Request
    ✓ Response 243ms
  #6
    ✓ Request
    ✓ Response 6ms

HTTP/1.1 200
Content-Type: application/json; charset=UTF-8
Transfer-Encoding: chunked
Date: Fri, 09 Jan 2026 13:39:05 GMT

Response file saved.
> 2026-01-09T163905.200.json

Response code: 200; Time: 255ms (255 ms); Content length: 471 bytes (471 B)
```

```
HTTP Requests 600ms ✓ Stopped. 6 tests passed 6 / 12 tests, 600ms
  1. GET /api/translatic 255ms
    ✓ Request
    ✓ Response 255ms
  #2
    ✓ Request
    ✓ Response 78ms
  #3
    ✓ Request
    ✓ Response 5ms
  #4
    ✓ Request
    ✓ Response 13ms
  #5
    ✓ Request
    ✓ Response 243ms
  #6
    ✓ Request
    ✓ Response 6ms

POST http://localhost:8080/api/translations
Content-Type: application/json

{
  "sourceText": "Hello world",
  "targetText": "Привет мир",
  "connections": [
    { "description": "First connection" }
  ]
}

HTTP/1.1 201
Content-Type: application/json; charset=UTF-8
Transfer-Encoding: chunked
Date: Fri, 09 Jan 2026 13:39:05 GMT

Response file saved.
> 2026-01-09T163905.201.json

Response code: 201; Time: 78ms (78 ms); Content length: 119 bytes (119 B)
```

```
HTTP Requests 600ms
  ✓ 1. GET /api/translate 255ms
    ✓ Request
    ✓ Response 255ms
  ✓ #2 78ms
    ✓ Request
    ✓ Response 78ms
  > ✓ #3 5ms
  > ✓ #4 13 ms
  ✓ #5 243 ms
    ✓ Request
    ✓ Response 243 ms
  > ✓ #6 6ms

✓ Stopped. 6 tests passed 6 / 12 tests, 600ms
POST http://localhost:8080/api/translations
Content-Type: application/json

{
  "targetText": "Only target"
}

###

HTTP/1.1 400
Content-Length: 0
Date: Fri, 09 Jan 2026 13:39:05 GMT
Connection: close

<Response body is empty>

Response code: 400; Time: 5ms (5 ms); Content length: 0 bytes (0 B)
```

```
HTTP Requests 600ms
  ✓ 1. GET /api/translate 255ms
    ✓ Request
    ✓ Response 255ms
  ✓ #2 78ms
    ✓ Request
    ✓ Response 78ms
  > ✓ #3 5ms
  > ✓ #4 13 ms
    ✓ Request
    ✓ Response 13 ms
  ✓ #5 243 ms
    ✓ Request
    ✓ Response 243 ms
  > ✓ #6 6ms

✓ Stopped. 6 tests passed 6 / 12 tests, 600ms
GET http://localhost:8080/api/translations/2
###

HTTP/1.1 200
Content-Type: application/json; charset=UTF-8
Transfer-Encoding: chunked
Date: Fri, 09 Jan 2026 13:39:05 GMT

Response file saved.
> 2026-01-09T163905-1.200.json

Response code: 200; Time: 13ms (13 ms); Content length: 174 bytes (174 B)
```

35:3 CRLF UTF-8 4 spaces

```
HTTP Requests 600ms
  ✓ 1. GET /api/translate 255ms
    ✓ Request
    ✓ Response 255ms
  ✓ #2 78ms
    ✓ Request
    ✓ Response 78ms
  > ✓ #3 5ms
  ✓ #4 13ms
    ✓ Request
    ✓ Response 13ms
  ✓ #5 243ms
    ✓ Request
    ✓ Response 243ms
  > ✓ #6 6ms

Stopped. 6 tests passed 6 / 12 tests, 600ms
DELETE http://localhost:8080/api/translations/1
###

HTTP/1.1 404
Content-Length: 0
Date: Fri, 09 Jan 2026 13:39:05 GMT

<Response body is empty>

Response code: 404; Time: 243ms (243 ms); Content length: 0 bytes (0 B)

35:3 CRLF UTF-8 4 spaces
```

```
HTTP Requests 600ms
  ✓ 1. GET /api/translate 255ms
    ✓ Request
    ✓ Response 255ms
  ✓ #2 78ms
    ✓ Request
    ✓ Response 78ms
  > ✓ #3 5ms
  ✓ #4 13ms
    ✓ Request
    ✓ Response 13ms
  ✓ #5 243ms
    ✓ Request
    ✓ Response 243ms
  > ✓ #6 6ms

Stopped. 6 tests passed 6 / 12 tests, 600ms
DELETE http://localhost:8080/api/translations/1
###

HTTP/1.1 404
Content-Length: 0
Date: Fri, 09 Jan 2026 13:39:05 GMT

<Response body is empty>

Response code: 404; Time: 6ms (6 ms); Content length: 0 bytes (0 B)
```

Тестирование с нагрузкой

Сервер был протестирован под симулированной нагрузкой с использованием *gatling*. Испытана нагрузка на сценариях, где в течение минуты или более отправляется 1 тысяча, 5 тысяч и 10 тысяч POST-запросов на */api/translations* в секунду.

- 1к POST-запросов:

---- Global Information -----		---Total---		----OK----	
KO-----					
> request count		60,000	60,000	-	-
> min response time (ms)		0	0	-	-
> max response time (ms)		267	267	-	-
> mean response time (ms)		2	2	-	-
> response time std deviation (ms)		9	9	-	-
> response time 50th percentile (ms)		1	1	-	-
> response time 75th percentile (ms)		1	1	-	-
> response time 95th percentile (ms)		1	1	-	-
> response time 99th percentile (ms)		2	2	-	-
> mean throughput (rps)		1,000	1,000	-	-
---- Response Time Distribution -----					
> OK: t < 800 ms		60,000 (100%)			
> OK: 800 ms <= t < 1200 ms		0 (0%)			
> OK: t >= 1200 ms		0 (0%)			
> KO		0 (0%)			

- 5к POST-запросов:

---- Global Information -----		---Total---		----OK----	
KO-----					

> request count	300,000 300,000 -
> min response time (ms)	0 0 -
> max response time (ms)	5,425 5,425 -
> mean response time (ms)	162 162 -
> response time std deviation (ms)	532 532 -
> response time 50th percentile (ms)	2 2 -
> response time 75th percentile (ms)	2 2 -
> response time 95th percentile (ms)	1,322 1,323 -
> response time 99th percentile (ms)	4,223 4,240 -
> mean throughput (rps)	4,918.03 4,918.03 -

---- Response Time Distribution -----

> OK: t < 800 ms	282,126 (94.04%)
> OK: 800 ms <= t < 1200 ms	1,275 (0.43%)
> OK: t >= 1200 ms	16,599 (5.53%)
> KO	0 (0%)

- 10к POST-запросов:

---- Global Information -----|---Total---|---OK---|---KO---

> request count	600,000 268,316 331,684
> min response time (ms)	3 301 3
> max response time (ms)	91,182 82,066 91,182
> mean response time (ms)	12,042 13,366 10,971
> response time std deviation (ms)	11,173 8,861 12,639
> response time 50th percentile (ms)	9,463 11,284 7,711
> response time 75th percentile (ms)	16,419 18,420 14,489
> response time 95th percentile (ms)	29,317 29,823 28,248

> response time 99th percentile (ms)		62,624 39,025 71,010
> mean throughput (rps)		4,477.61 2,002.36 2,475.25
---- Response Time Distribution -----		
> OK: t < 800 ms		1,974 (0.33%)
> OK: 800 ms <= t < 1200 ms		2,326 (0.39%)
> OK: t >= 1200 ms		264,016 (44%)
> KO		331,684 (55.28%)
---- Errors -----		
> j.n.ConnectException: connect(..) failed: Cannot assign requested address (92.17%)		305,715
> i.n.c.ConnectTimeoutException: connection timed out after 10000 ms: /127.0.0.1:8080 16,015 (4.83%)		
> Request timeout to 127.0.0.1/127.0.0.1:8080 after 60000 ms		9,943 (3%)
> j.i.IOException: Premature close		11 (0%)
=====		

Результаты показывают, что нагрузку в 5 тысяч запросов/с и ниже сервер выдерживает без особых трудностей, подавляющее большинство таких запросов получают ответ меньше, чем за секунду. Когда нагрузка достигает 10 тысяч запросов/с - больше половины запросов не получают ответ (92% из них отбрасывает операционная система из-за длинной очереди), а средняя скорость обработки увеличивается до 12 секунд. Учитывая, что даже 1 тысяча запросов в секунду это нереалистично высокая мерка для потенциального пользования данной программы - за её масштабируемость беспокоиться нет необходимости.

Сборка и запуск

Для сборки клиента используется утилита *cmake*:

```
cmake -S Woofie -B build
cmake --build build
```

Для упрощения сборки вне контейнера (в частности для подгрузки библиотек и среды разработки QtCreator) можно применить менеджер пакетов и утилиты *nix*:

```
nix develop
```

Предусмотрена сборка через *docker*:

```
docker build -t buildimage
```

В отличие от серверной стороны, ввиду того, что клиент является GUI-приложением, его запуск изнутри контейнера затруднён, если не невозможен.

Для ручного тестирования, сборки и запуска сервера используется *gradle*:

```
gradle test
```

```
gradle build  
gradle bootRun
```

Для ручного запуска (пустой) базы данных PostgreSQL так же применяется *nix* :
`nix run`

Базу данных и сервер можно собрать и запустить используя *docker-compose*
`docker-compose up`