

## “Разработка библиотеки на языке Rust для работы с форматом данных Transit”

Фомин Е. М., Университет ИТМО, Санкт-Петербург

Научный руководитель: Радченко Ирина Алексеевна, Университет ИТМО, Санкт-Петербург.

### Введение

Разработка программного продукта с применением микросервисной или клиент-серверной архитектур подразумевает интенсивный обмен данными между приложениями. В сравнении с интерфейсами, предоставляемыми веб-сервисами, предметная область отдельно взятых микросервисов довольно локальна и в большей степени опирается на какие-либо соглашения, например, технические детали, заложенные в реализацию взаимодействующих приложений, никак не отраженные в данных, которыми они обмениваются.

Чтобы избежать использования схем или создания собственного формата данных, разработчику предлагается выбор из множества существующих форматов, ориентированных на взаимодействие между приложениями и с включенными механизмами расширения для предметной области. Одним из таких решений является формат Transit. Его главное отличие — работа поверх JSON, что позволяет переиспользовать существующие парсеры — это особенно удобно в браузере.

Для разработки приложений, к которым предъявляются особые требования по производительности, зачастую используют системные языки программирования, одним из которых является язык Rust. Относительно новый, но позитивно принятый в индустрии, язык плавно внедряется, в том числе в виде микросервисов, компилируется в WebAssembly и применяется во встраиваемых системах. Реализация библиотеки для поддержки формата Transit для Rust позволит приложениям, работающим на множестве платформ, обмениваться данными в формате, разработанном специально для этой цели и с поддержкой множества других языков программирования.

### Цель работы

Разработать библиотеку для языка Rust, позволяющую написанным на этом языке приложениям работать с данными в формате Transit.

### Базовые положения работы

Наиболее зрелым, распространенным и фактически стандартным решением по сериализации и десериализации данных в языке Rust является фреймворк Serde. Использование промежуточной модели данных, именуемой Serde Data Model, позволяет сериализовать структуру данных языка, для которой определено представление в SDM, в любой из поддерживаемых фреймворком форматов.

В случае с форматом Transit данные в промежуточной модели будут выглядеть иначе, нежели при сериализации в, например, YAML. Для наглядности: тип данных множества будет упрощен до массива, а Transit хранит эту информацию, при этом упрощенное представление структуры данных может быть удобно для внешнего API, но неприемлемо для внутреннего использования, поэтому требуется поддержать оба представления в SDM одновременно.

### Промежуточные результаты

В рамках данной работы:

- Изучены возможности других форматов обмена данными: MessagePack, Ion, Protobuf
- Изучены способы расширения функционала фреймворка Serde
- Реализована сериализация для базовых типов

- Проработаны возможности сериализации в форматы, отличные от Transit
- Реализована сериализация типов-расширений

**Основной результат**

В результате работы разработана библиотека для работы с форматом Transit из программ на языке Rust.