

Выполнил(а) Барсуков М. А., № группы P3115, оценка
Фамилия И.О. студента не заполнять

Название статьи/главы книги/видеолекции Protobuf — What & Why?		
ФИО автора статьи (или e-mail) Swaminathan Muthuveerappan	Дата публикации (не старше 2019 года) "08" июля 2021 г.	Размер статьи (от 400 слов) 2092
Прямая полная ссылка на источник или сокращённая ссылка (bit.ly, tr.im и т.п.) https://medium.com/nerd-for-tech/protobuf-what-why-fcb324a64564		
Теги, ключевые слова или словосочетания Protobuf; Двоичные форматы данных; Сериализация		
Перечень фактов, упомянутых в статье <ol style="list-style-type: none"> 1. Protobuf – созданный в Google бинарный формат данных. 2. Protobuf обычно используется для передачи данных между микросервисами. 3. Protobuf используется во множестве высоконагруженных приложениях: многих сервисов Amazon, Google и т.д. 4. Поскольку сообщение Protobuf находится в двоичном формате, для его передачи требуется меньше места и пропускной способности. 5. Сообщения Protobuf сериализуются в двоичный формат, а не в текст, такой как JSON, поэтому сообщения в Protobuf не человекочитаемы. 6. Для преобразования доменного объекта в двоичный файл используется proto файл с определением структуры сообщения с использованием особого синтаксиса. 7. Благодаря схеме сообщения Protobuf не содержат избыточных метаданных для структуры. 8. Каждому полю в сообщении Protobuf присваивается уникальный номер. 9. Proto файлы управляются из Schema Registry, в котором хранятся определения схем. 10. Двоичный файл сообщения Protobuf можно зашифровать/расшифровать и вручную. 		
Позитивные следствия и/или достоинства описанной в статье технологии (минимум три пункта) <ol style="list-style-type: none"> 1. Сериализация и десериализация Protobuf выполняется быстрее, чем у JSON или XML. 2. Размер сообщения в Protobuf намного меньше аналогичного для простых текстовых форматов. 3. Используя уникальный номер поля, Protobuf обеспечивает превосходную обратную совместимость. 4. Protobuf проверяет тип значения заданных данных во время кодирования и декодирования, обеспечивая целостность данных во время передачи данных. 5. У Protobuf есть автогенератор документации по созданным структурам. 		
Негативные следствия и/или недостатки описанной в статье технологии (минимум три пункта) <ol style="list-style-type: none"> 1. Не декодированные данные Protobuf неудобочитаемы (так как это двоичный формат), тогда как JSON можно спокойно понять без десериализации. 2. Protobuf требует привязку модели данных к схеме, что не всегда может быть удобно. 3. Использование Protobuf вне коммуникации между сервисами довольно ограничено, тогда как JSON всё ещё остается самым популярным решением для различных API. 4. Protobuf сложнее отлаживать из-за того, что это двоичный формат. 5. Использовать JSON намного удобнее, когда клиентом является веб-браузер или приложение на Node.js. 		
Ваши замечания, пожелания преподавателю или анекдоты о программистах <ol style="list-style-type: none"> 1) «Ищется Ruby on Rails разработчик, строго без опыта PHP» (анекдот от 2008) 2) Вы слышали о новом суперкомпьютере Cray? Он настолько быстрый, что выполняет бесконечный цикл за 6 секунд. 3) WYSIWYMGYRRLAAGW: What You See Is What You Might Get If You're Really Really Lucky And All Goes Well. 		
P.S.: За эту неделю я больше не нашел, все остальные шутки даже более бородастые, чем эти :(