
gRPC

protobuf
Protocol Buffers

GRPC

What are protocol buffers?

- Uses **Interface Description Language** as a flexible, efficient, automated mechanism for **serializing structured data**
  - similar to XML (or JSON), but smaller, faster, and simpler
  - Google specifically created the protocol internally in the early 2000s to be more efficient than XML for similar tasks

- Define the **structure once**. . . then use **special generated source code** to easily write and read your structured data to and from a variety of data streams and using a variety of languages

https://developers.google.com/protocol-buffers/

protobuf
Protocol Buffers

GRPC

Why not just use XML?

Protocol buffers:

- are simpler
- are 3 to 10 times smaller
- are 20 to 100 times faster
- are less ambiguous
- generate data access classes that are easier to use programmatically

```
<person>
  <name>John Doe</name>
  <email>jdoe@example.com</email>
</person>
```

```
# Textual representation of a protocol buffer.
# This is *not* the binary format used on the wire.

person {
  name: "John Doe"
  email: "jdoe@example.com"
}
```

When this message is encoded to the protocol buffer binary format. . . it would probably be 28 bytes long and take around 100-200 nanoseconds to parse. The XML version is at least 69 bytes if you remove whitespace, and would take around 5,000-10,000 nanoseconds to parse.

https://developers.google.com/protocol-buffers/docs/overview

protobuf
Protocol Buffers

GRPC

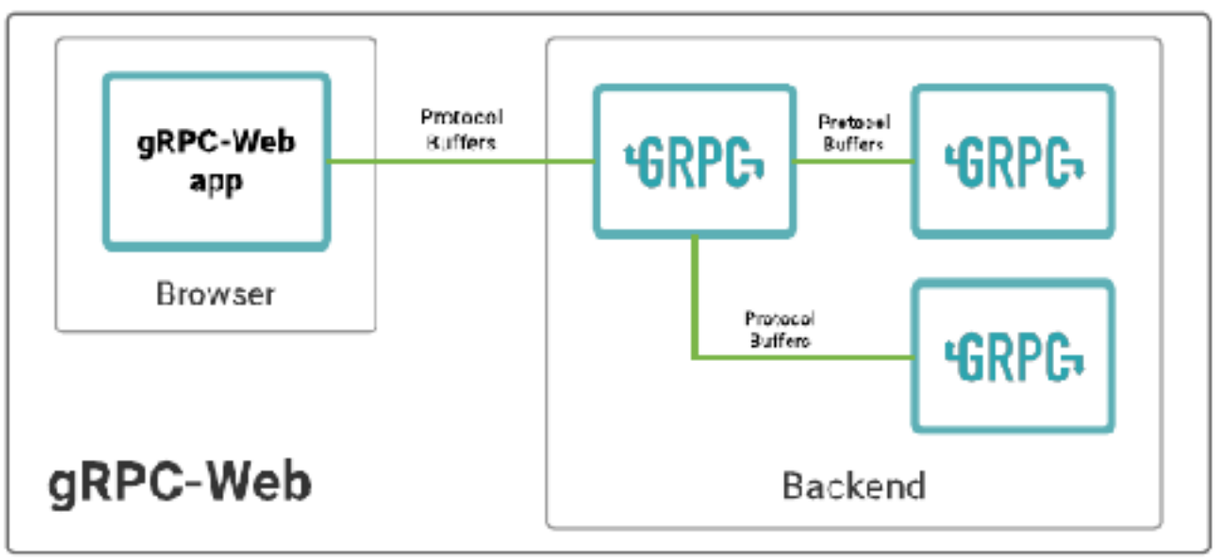
```
message Data {
  int64 IntValue = 1;
  string StringValue = 2;
}
```

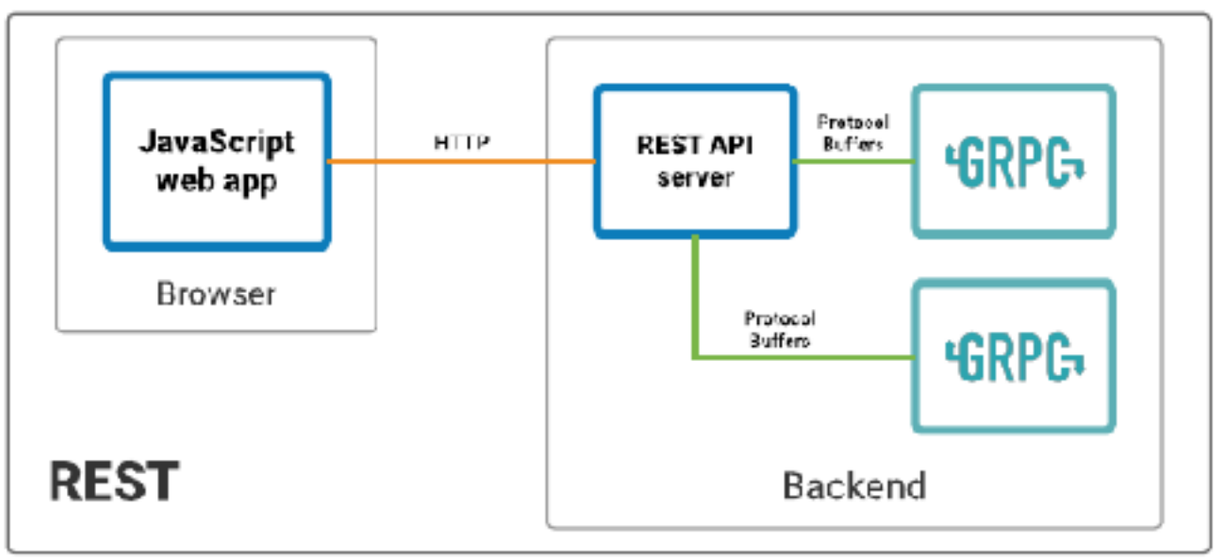
key (field #, wire type) : value

Type	Meaning	Used For
0	Varint	int32, int64, uint32, uint64, sint32, sint64, bool, enum
1	64-bit	fixed64, sfixed64, double
2	Length-delimited	string, bytes, embedded messages, packed repeated fields
3	Start group	groups (deprecated)
4	End group	groups (deprecated)
5	32-bit	fixed32, sfixed32, float

https://developers.google.com/protocol-buffers/docs/encoding

New feature (2018): gRPC-Web





https://grpc.io/blog/grpc-web-ga

Prior API solution:
- Use an rpc option to expose API endpoints from gRPC
- API endpoints will use HTTP1.1 (REST/JSON)
- However, both the HTTP1.1 API and gRPC HTTP/2 interface use a **single TCP port**

- https://grpc.io/blog/coreos
- https://cloud.google.com/endpoints/docs/grpc-service-config/reference/rpc/google.api