Subject: Transfer compress data
Posted by forlano on Wed, 26 Apr 2023 10:59:38 GMT
View Forum Message <> Reply to Message

Hello,

my app communicate with a MySql database online.
I wonder if the data transfer can be done in a compressed/zipped way to speed the whole procedure.

Thanks,
Luigi

Subject: Re: Transfer compress data
Posted by zsolt on Fri, 28 Apr 2023 22:25:59 GMT
View Forum Message <> Reply to Message

Hello Luigi,

A had a same problem, so I created a backend part for my U++ software.
It receives JSON data as arguments in HTTP and the responses are also in JSON. This way it can be used by not only my GUI app, but a Javascript SPA (single page app) also.
HTTP is good, because I put my backent behind an Nginx reverse proxy. Nginx can compress the responses.

The other problem, you can had solved with this solution is the latency, caused by multiple SQL queries. E.g. user opens a data entry window with a lot of droplists and/or converts, based on the database, that generates a lot of SQL queries one by one, adding a very large latency. Using this JSON requests, you can send all the parameters to the server (the backend) in a JSON. It runs all the queries in place and sends back all the results in one JSON, so the latency is one request-response time only.

Jsonize framework is very useful here.

Subject: Re: Transfer compress data
Posted by forlano on Sat, 29 Apr 2023 13:01:20 GMT
View Forum Message <> Reply to Message

zsolt wrote on Sat, 29 April 2023 00:25

Jsonize framework is very useful here.

Hello zsolt,

thank you for the reply. I'll move in this direction too.

Best regards,
Luigi

Subject: Re: Transfer compress data
Posted by zsolt on Sat, 29 Apr 2023 16:24:01 GMT
View Forum Message <> Reply to Message

It is good to know, that if you don't plan using a Javascript client, the normal Serialize framework can be much better (faster and more compact).
JSON is not very efficient to transfer query results (2 dimension array of values), so I had to create a class to implement it more efficiently.
It serializes field names as an array of strings and rows as raw JSON value array (2 dimensions). Something like a CSV.