**Streaming data in gRPC**

Source project: [aieos Compose 1.46 - Edit Project (27) (cd8.io)](https://dant-playground.compose.forge.cd.int.cd8.io/Projects/Project/27)

Data file to test: flax\_model.msgpack in -

https://dev.azure.com/clrdyn/DataScience/\_git/PyORM?path=/resources/model\_files

gRPC has a 4BM limit on messages request so, if we need to send a data over 4MB, we need to split the data and stream it.

**[C# Side]**

* [Call gRPC services with the .NET client | Microsoft Learn](https://learn.microsoft.com/en-us/aspnet/core/grpc/client?view=aspnetcore-7.0#client-streaming-call)

Source: DataConfig\_DeserializeModel in ConfigCustonAction.cs

Instead of calling this:

response = await client.StartTrainingAsync(request);

Create a client call:

call = client.DeserializeSherlockModel();

And stream the data batch inside of for loop:

call.RequestStream.WriteAsync(request);

When the client has finished sending messages, CompleteAsync() should be called to notify the service.

await call.RequestStream.WriteAsync(request);

**[Python Side]**

[Basics tutorial | Python | gRPC](https://grpc.io/docs/languages/python/basics/#request-streaming-rpc)

Source: DeserializeSherlockModel in server.py

* **Check Input Stream on Compose**

**Graphical user interface, application, Teams

Description automatically generated**

This will add stream in MicroService.proto?

rpc DeserializeSherlockModel(**stream** DeserializeSherlockModelRequest) returns (DeserializeSherlockModelResponse) {}

* Use request\_iterator and returns a single response value.

for batch in request\_iterator:

   \_serialized = \_serialized + batch.SerializedModel

‘’’

return ServiceTypes.DeserializeSherlockModelResponse(Success=res\_bool)