syntax = "proto3";

package tensorflow.serving;

import "tensorflow/core/protobuf/config.proto";

import "tensorflow/core/protobuf/named\_tensor.proto";

import "tensorflow\_serving/apis/model.proto";

option cc\_enable\_arenas = true;

message SessionRunRequest {

// Model Specification. If version is not specified, will use the latest

// (numerical) version.

ModelSpec model\_spec = 1;

// Tensors to be fed in the step. Each feed is a named tensor.

repeated NamedTensorProto feed = 2;

// Fetches. A list of tensor names. The caller expects a tensor to

// be returned for each fetch[i] (see RunResponse.tensor). The

// order of specified fetches does not change the execution order.

repeated string fetch = 3;

// Target Nodes. A list of node names. The named nodes will be run

// to but their outputs will not be fetched.

repeated string target = 4;

// If true, treat names in feed/fetch/target as alias names than actual tensor

// names (that appear in the TF graph). Alias names are resolved to actual

// names using `SignatureDef` in SavedModel associated with the model.

bool tensor\_name\_is\_alias = 6;

// Options for the run call. \*\*Currently ignored.\*\*

RunOptions options = 5;

}

message SessionRunResponse {

// Effective Model Specification used for session run.

ModelSpec model\_spec = 3;

// NOTE: The order of the returned tensors may or may not match

// the fetch order specified in RunRequest.

repeated NamedTensorProto tensor = 1;

// Returned metadata if requested in the options.

RunMetadata metadata = 2;

}

// SessionService defines a service with which a client can interact to execute

// Tensorflow model inference. The SessionService::SessionRun method is similar

// to MasterService::RunStep of Tensorflow, except that all sessions are ready

// to run, and you request a specific model/session with ModelSpec.

service SessionService {

// Runs inference of a given model.

rpc SessionRun(SessionRunRequest) returns (SessionRunResponse);

}