// Input used in serving APIs. Based on the tensorflow.Example family of

// feature representations.

syntax = "proto3";

option cc\_enable\_arenas = true;

import "tensorflow/core/example/example.proto";

package tensorflow.serving;

// Specifies one or more fully independent input Examples.

// See examples at:

// https://github.com/tensorflow/tensorflow/blob/master/tensorflow/core/example/example.proto

message ExampleList {

repeated tensorflow.Example examples = 1;

}

// Specifies one or more independent input Examples, with a common context

// Example.

//

// The common use case for context is to cleanly and optimally specify some

// features that are common across multiple examples.

//

// See example below with a search query as the context and multiple restaurants

// to perform some inference on.

//

// context: {

// features: {

// feature: {

// key : "query"

// value: {

// bytes\_list: {

// value: [ "pizza" ]

// }

// }

// }

// }

// }

// examples: {

// features: {

// feature: {

// key : "cuisine"

// value: {

// bytes\_list: {

// value: [ "Pizzeria" ]

// }

// }

// }

// }

// }

// examples: {

// features: {

// feature: {

// key : "cuisine"

// value: {

// bytes\_list: {

// value: [ "Taqueria" ]

// }

// }

// }

// }

// }

//

// Implementations of ExampleListWithContext merge the context Example into each

// of the Examples. Note that feature keys must not be duplicated between the

// Examples and context Example, or the behavior is undefined.

//

// See also:

// tensorflow/core/example/example.proto

// https://developers.google.com/protocol-buffers/docs/proto3#maps

message ExampleListWithContext {

repeated tensorflow.Example examples = 1;

tensorflow.Example context = 2;

}

message Input {

oneof kind {

ExampleList example\_list = 1 [lazy = true];

ExampleListWithContext example\_list\_with\_context = 2 [lazy = true];

}

}