# tf.parse_single_sequence_example

```
parse_single_sequence_example(
    serialized,
    context_features=None,
    sequence_features=None,
    example_name=None,
    name=None
)
```

Defined in `tensorflow/python/ops/parsing_ops.py`.

Parses a single `SequenceExample` proto.

Parses a single serialized `SequenceExample` proto given in `serialized`.

This op parses a serialized sequence example into a tuple of dictionaries mapping keys to `Tensor` and `SparseTensor` objects respectively. The first dictionary contains mappings for keys appearing in `context_features`, and the second dictionary contains mappings for keys appearing in `sequence_features`.

At least one of `context_features` and `sequence_features` must be provided and non-empty.

The `context_features` keys are associated with a `SequenceExample` as a whole, independent of time / frame. In contrast, the `sequence_features` keys provide a way to access variable-length data within the `FeatureList` section of the `SequenceExample` proto. While the shapes of `context_features` values are fixed with respect to frame, the frame dimension (the first dimension) of `sequence_features` values may vary between `SequenceExample` protos, and even between `feature_list` keys within the same `SequenceExample`.

`context_features` contains `VarLenFeature` and `FixedLenFeature` objects. Each `VarLenFeature` is mapped to a `SparseTensor`, and each `FixedLenFeature` is mapped to a `Tensor`, of the specified type, shape, and default value.

`sequence_features` contains `VarLenFeature` and `FixedLenSequenceFeature` objects. Each `VarLenFeature` is mapped to a `SparseTensor`, and each `FixedLenSequenceFeature` is mapped to a `Tensor`, each of the specified type. The shape will be `(T,) + df.dense_shape` for `FixedLenSequenceFeature` `df`, where `T` is the length of the associated `FeatureList` in the `SequenceExample`. For instance, `FixedLenSequenceFeature([])` yields a scalar 1-D `Tensor` of static shape `[None]` and dynamic shape `[T]`, while `FixedLenSequenceFeature([k])` (for `int k >= 1`) yields a 2-D matrix `Tensor` of static shape `[None, k]` and dynamic shape `[T, k]`.

Each `SparseTensor` corresponding to `sequence_features` represents a ragged vector. Its indices are `[time, index]`, where `time` is the `FeatureList` entry and `index` is the value's index in the list of values associated with that time.

`FixedLenFeature` entries with a `default_value` and `FixedLenSequenceFeature` entries with `allow_missing=True` are optional; otherwise, we will fail if that `Feature` or `FeatureList` is missing from any example in `serialized`.

`example_name` may contain a descriptive name for the corresponding serialized proto. This may be useful for debugging purposes, but it has no effect on the output. If not `None`, `example_name` must be a scalar.

## Args:

- `serialized`: A scalar (0-D Tensor) of type string, a single binary serialized `SequenceExample` proto.
- `context_features`: A `dict` mapping feature keys to `FixedLenFeature` or `VarLenFeature` values. These features are associated with a `SequenceExample` as a whole.

- `sequence_features`: A `dict` mapping feature keys to `FixedLenSequenceFeature` or `VarLenFeature` values. These features are associated with data within the `FeatureList` section of the `SequenceExample` proto.
- `example_name`: A scalar (0-D Tensor) of strings (optional), the name of the serialized proto.
- `name`: A name for this operation (optional).

## Returns:

A tuple of two `dict`s, each mapping keys to `Tensor`s and `SparseTensor`s. The first dict contains the context key/values. The second dict contains the feature_list key/values.

## Raises:

- `ValueError`: if any feature is invalid.

---