bitpack Documentation

Release 0.1

Outernet Inc

April 22, 2016

Contents

Data serialization library, essentially turning sequences of data structures into a compact binary representation.

CHAPTER 1

Source code

bitpack source code can be found on GitHub and is released under GPLv3 license. See the `COPYING` file in the source tree for more information.

CHAPTER 2

Documentation

2.1 Working with bitpack

This section gives you a quick overview of bitpack library usage.

2.2 API documentation

class `bitpack.`**`BitField`**(*name*, *index*, *width*, *data_type*)
Bases: `object`

A class representing a single field within a `BitStream`. It handles the serialization / deserialization of the values / data that is passed to it. The following constructor parameters are available:

Parameters

- **`name`** – the name of the field as it was declared
- **`index`** – integer, used to keep the order of fields as declared
- **`width`** – integer, the needed bit-width for the data
- **`data_type`** – unique identifier of the data type for which there exists a registered serializer / deserializer

`data_type`
Returns the data type of the field that was specified in the field declaration.

`deserialize`(*bits*)
Perform deserialization of the passed in data and return it in it's deserialized form.

Parameters **`bits`** – data to be deserialized

`name`
Returns the name of the field by which it was declared on the `BitStream` class.

classmethod **`register_data_type`**(*data_type*, *serializer_fn*, *deserializer_fn*)
Add a new data serializer and deserializer to all `BitField` objects (including subclasses as well).

Parameters

- **`data_type`** – name of the type
- **`serializer_fn`** – function that performs serialization
- **`deserializer_fn`** – function that performs deserialization

serialize(*value*)
Perform serialization of the passed in value and return it in it's serialized form.

Parameters **value** – value to be serialized

width
Returns the bit-width of the field that was specified in the field declaration.

class bitpack.**BitStream**(*data*)
Bases: object

Expected to be subclasses and fields declared on subclasses that define in what way should the data be serialized and deserialized.

Overridable attributes:

Attr start_marker A string used to indicate the start of a data record in it's serialized form.

Attr end_marker A string used to indicate the end of a data record in it's serialized form.

Constructor arguments:

Parameters **data** – it serves multiple purposes: - as a string it represents the data to be deserialized - as an iterable of dicts it's the source data to be

serialized

deserialize()
Perform deserialization of the data that was passed to the constructor and return it in it's deserialized form.

end_marker = None

classmethod **from_bytes**(*raw_bytes*)
Helper method to instantiate a class with the passed in raw_bytes and implicitly call and return the result of it's deserialize method.

Parameters **raw_bytes** – data to be deserialized

serialize()
Perform serialization of the data that was passed to the constructor and return it in it's serialized form.

start_marker = None

classmethod **to_bytes**(*raw_data*)
Helper method to instantiate a class with the passed in raw_data and implicitly call and return the result of it's serialize method.

Parameters **raw_data** – data to be serialized

bitpack.**register_data_type**(*data_type*, *serializer_fn*, *deserializer_fn*)
Add a new data serializer and deserializer to all BitField objects (including subclasses as well). This is just a helper function that simply delegates calls to the classmethod on BitField itself.

Parameters

- **data_type** – name of the type
- **serializer_fn** – function that performs serialization
- **deserializer_fn** – function that performs deserialization

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

b

B

D

E

F

N

R

S

T

W