Difference between storing an ObjectId and its string form, in MongoDB

I'm a little confused by Mongo DB's use of ObjectIds. Sure they're great for creating IDs client-side that almost definitely don't conflict with other client-side created ids. But mongo seems to store them in some special way. Storing a string representation of the id is *different* from storing an object id as an object. Why is this?

Doesn't the string form have all the same information that the object form has? Why does mongo go to such lengths to differentiate those two forms? It screws me up when I try to compare _ids sent from a frontend for example. My database is in no way consistent with whether it stores string-form ids or object-form ids, and tho my code is certainly partially to blame, I mostly blame mongo for making this so weird.

Am I wrong that this is weird? Why does mongo do it this way?

mongodb objectid

asked Jan 12 '15 at 7:12
B T
23.3k 23 121 152

2 Answers

I, personally, blame your code. I get around this pefectly fine in my applications by coding the right way. I convert to string in code to compare and I ensure that anything that looks like an `ObjectId` is actually used as a `ObjectId` .

It is good to note that between the `ObjectId` (http://docs.mongodb.org/manual/reference/object-id/) and it's hex representation there is in fact 12 bytes of difference, the `ObjectId` being 12 bytes and it's hex representation being 24.

Not only is it about storage efficiency but also about indexes; not just because they are smaller but also since the `ObjectId` can be used in a special manner to ensure that only parts of the index are loaded; the parts that are used. This becomes most noticeable when inserting, where only the latest part of that index needs to be loaded in to ensure uniqueness. You cannot guarantee such behaviour with its hex representation.

I would strongly recommend you do not use the `OjbectId` 's hex representation. If you want to "make your life easier" you would be better off creating a different `_id` which is smaller but somehow just as unique and index friendly.

edited Jan 12 '15 at 20:50

answered Jan 12 '15 at 8:16
Sammaye
33.7k 6 61 111

You shouldn't need to convert ObjectIds to strings to compare them. They do have an equals method (at least in mongo native / mongoskin). – B T Jan 12 '15 at 20:43

The reference you have says that ObjectIds are 12 bytes, not 16 – B T Jan 12 '15 at 20:44

@BT whops my bad, must have been thinking of something else – Sammaye Jan 12 '15 at 20:50

@BT if the framework supports that then that brilliant, I use PHP personally – Sammaye Jan 12 '15 at 20:51

`ObjectId` is 12 bytes when it's stored internally, which is more compact than the hexadecimal string representation of it. The two are different things.

You can reflow your whole DB and use a uniform `_id` field to solve this and make sure your code saves in the same format. `ObjectId` are fast to generate by MongoDB so I'd use that when creating new documents.

answered Jan 12 '15 at 7:21
bakkal
37.8k 9 73 75

Ah so its all about storage efficiency then? All my _ids are ObjectIds, but lots of reference fields aren't right now, I suppose I should fix things up – B T Jan 12 '15 at 7:28

1 Field size also might affect the indexing/querying performance (not sure by how much for `ObjectId` vs `ObjectId.str` , but MongoDB was probably built/tested with ObjectId so let's use that :D) – bakkal Jan 12 '15 at 7:35

This is actually pretty annoying since there's no real good way to serialize object ids. Any requests that come from the client have to be explicitly combed to replace string ids with object ids. Is there a way around that? – B T Jan 12 '15 at 7:41

Depends on your web stack/framework since it involves the client, so I'd post a separate question with the details – bakkal Jan 12 '15 at 7:48

Theoretically, couldn't mongo (client or server) detect when a string is a valid ObjectId hex value and automatically convert it? Then the user wouldn't need to differentiate, and the only overhead would be a minor deserialization/serialization cost – B T Jan 12 '15 at 20:46