It's been a long time we haven't seen news on this blog. It is time to fix it!

About Storages

New connector for Amazon S3 added!

The `s3storage` is able to do i/o on the Amazon S3 servers as classical storage server. The storage description is:

```
{
"type": "s3",
"AWSIdentifier": <AWS Identifier; string>,
"password": <AWS Secret; string>,
"server": <Destination bucket; string>,
"url": <Destination url; string> (optional)
}
```

More and more on `indexstorage`!

The index storage is able to index documents when putting, getting, or removing them in a sub storage. The indices are used to query documents faster than a classical allDocs (using Complex Queries). An index is a document that contains only needed other document informations according to the storage description.

- Description + index database indexing way changed to make it faster
- It only uses index databases to search for documents
- We can use several index database documents
- Possibility to name the index database documents
- Possibility to store index databases documents where you want (using a storage description)
- Possibility to get the index database documents

The indexstorage description, now, looks like:

```
{
"type": "indexed",
"sub_storage": {
<storage description>
},
"indices": [{
"id": <Index document id>,
"index": [<metadata to index>, ...]
"sub_storage": {<storage description>} // optional, default: parent sub_storage
    // Why not to store the index database documents on localstorage to improve speed?
  }, {
...
}
}
```

Documentation is already up to date!

About Queries

Complex Queries is a standalone javascript module now, compatible with nodejs, requirejs, web workers and, of course, classical recent browsers.

- NodeJs

  ```
  var complex_queries = require('./complex_queries.js');
  ```

- RequireJs

  ```
  define(['complex_queries'], function (complex_queries) { .. })
  ```

- Web Workers

  ```
  importScripts('./complex_queries.js');
  console.log(complex_queries);
  ```

- Classical browsers

  ```
  <script src="complex_queries.js"></script>
  <script> console.log(complex_queries); </script>
  ```

The module provides a completly different API, but don't worry, all storages are already up to date. Don't hesitate to test the queries using the JIO dashboard or the query example web page on the official repository. Or see the documentation to see how to use complex queries with JIO.

We can also convert any Complex Queries to any other query type by overriding some query methods. These methods are used as events during the parsing process and to make convertion easier, and object is shared between all methods. Here is an example of converting a Complex Query to a human readable string.

The documentation to explain how to use Complex Queries inside a storage or outside JIO is comming soon.

Comming soon!

ERP5 external i/o thanks to erp5storage!.

This storage will be able to do some i/o on an ERP5 database. We already have some prototypes done with OfficeJS, but allDocs does not manage complex queries yet.

JIO Grows up!

We need you! Do not hesitate to contribute!

Thank you all!

Tristan Cavelier