

officejs.com

Unified Cloud Storage
with JIO

Cédric Le Ninivin – cedric.leninivin@nexedi.com

Suppose that a developer wants to develop a Javascript application that is compatible with most Cloud providers...

Cloud Storage APIs



A Developer Nightmare!

Suppose that the same developer wants to be compatible with most Content Management System...

CMS Storage APIs



A Developer Nightmare!

Suppose now that all this is possible by learning only one simple API...

JIO: One API for All Clouds

- jio.get
- jio.put
- jio.post
- jio.remove
- jio.allDocs

Based on CouchDB API
with extensions for queries

One API ↓ get, put, post, remove, allDocs

JIO				
Local	DAV	Mongo	S3	ERP5
XWiki	Drupal	CouchDB	Qiniu	...



Any API

JIO: Cloud, better than Cloud

- Encryption
- Splitting
- Indexing
- Online
- Offline
- Replication

Enterprise Grade
=
Trade Secret Protection
+
Performance Optimization

JIO Docs

<http://jio.readthedocs.org/en/latest/>



[Getting Started](#)

[Downloads](#)

[Support](#)

We have a [forum](#) and an [issue tracker](#) on GitHub.

[Stay Informed](#)

You can follow our [blog](#) to get notified about new releases.

[Quick search](#)

Welcome to jIO

jIO is a JavaScript library that allows to manage JSON documents on local or remote storages in asynchronous fashion. jIO is an abstracted API mapped after CouchDB, that offers connectors to multiple storages, special handlers to enhance functionality (replication, revisions, indexing) and a query module to retrieve documents and specific information across storage trees.

How does it work?

jIO is composed of two parts - jIO core and storage libraries. The core makes use of storage libraries (connectors) to interact with the associated remote storage servers. Some queries can be used on top of the `.allDocs()` method to query documents based on defined criteria.

jIO uses a job management system, so each method call adds a job into a queue. The queue is copied in the browser's local storage (by default), so it can be restored in case of browser crash. Jobs are invoked asynchronously and ongoing jobs are not able to re-trigger to prevent conflicts.

Copyright and license

jIO is open source and is licensed under the [LGPL](#) license.

Fork me on GitHub

v: latest ▼

Demo 1: OfficeJS

您的 App

JQuery Mobile

Chrome

Firefox

IE

...

JIO

Local

DAV

ERP5

Qiniu

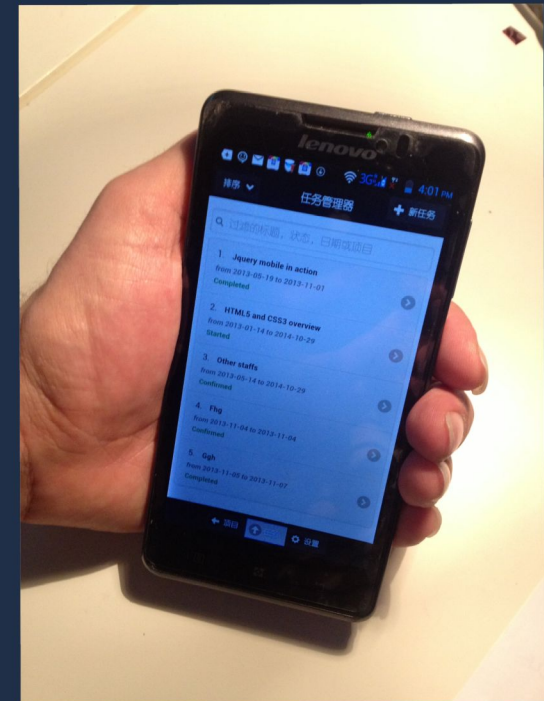
...

HTML5

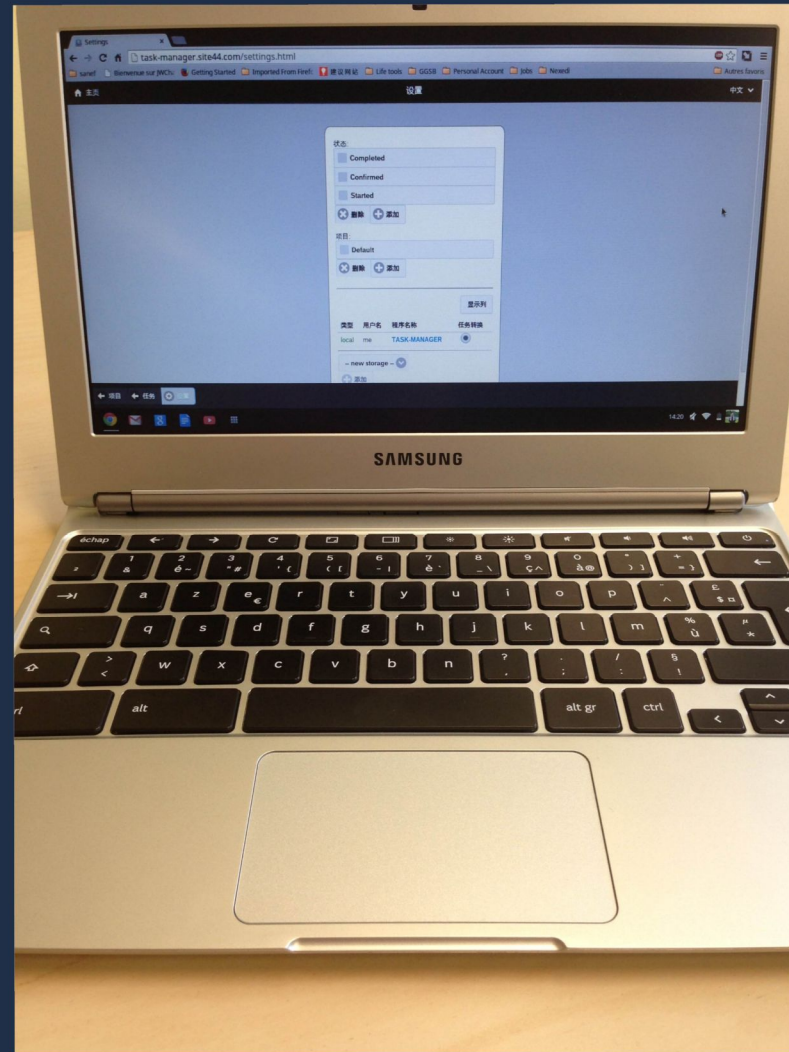
JIO-Local Storage on Android/iOS

Wifi SSID: officejs

Web: <http://10.0.0.200:12001/tc>



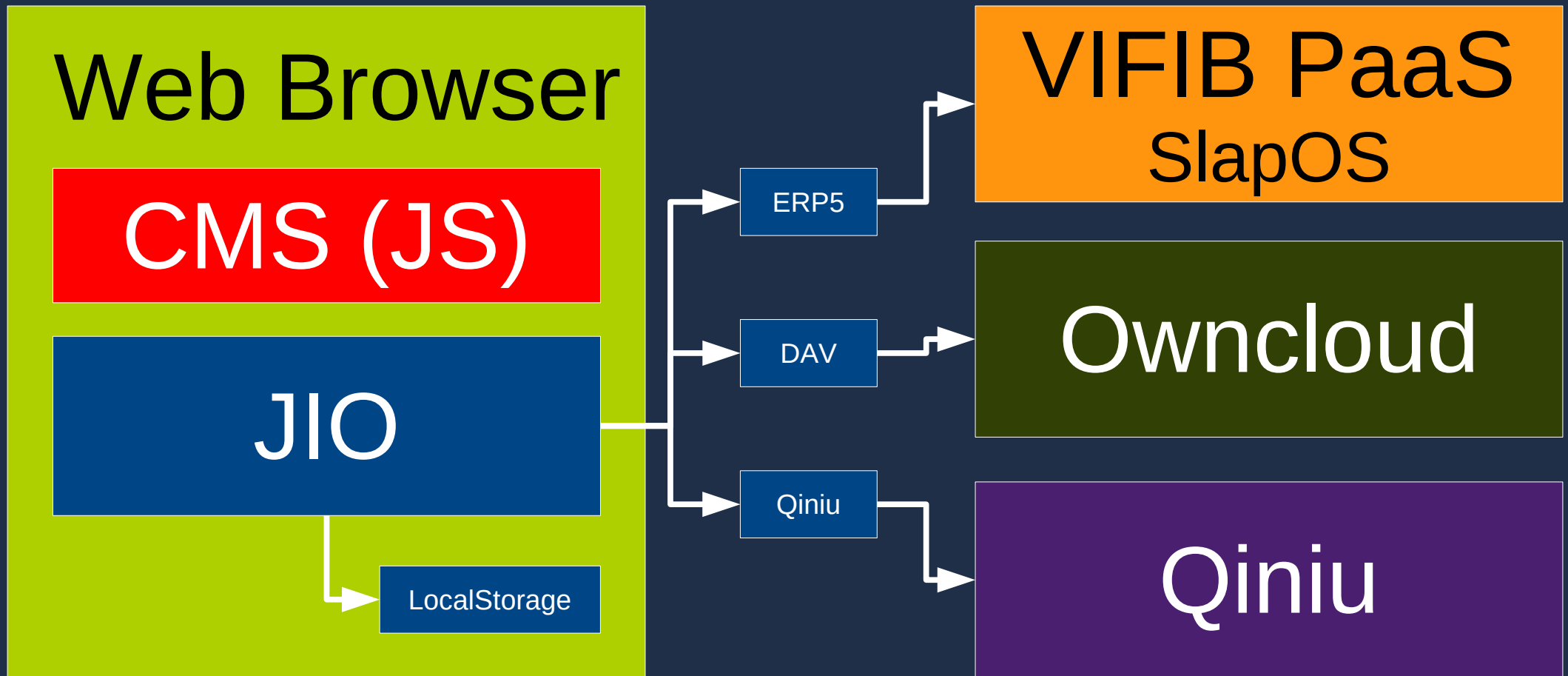
JIO-Qiniu on Desktop



Only One Source Code!

```
jio_project.post(  
  { "type": "Task Report",  
    "project": project,  
    "reference": uuid(),  
    "modified": new Date()  
  },  
  function (err, resp) {  
    // ...  
  }  
);
```

Demo 2: CMS in JS



Conclusion

- JIO
 - One simple API
 - All clouds
 - All countries
- Offline HTML5
 - Scalability at zero
 - All PC
 - All phones

¥10,000,000 saved by your startup

[http://jio.readthedocs.org/en/latest/
officejs.com/dev](http://jio.readthedocs.org/en/latest/officejs.com/dev)

The logo for officejs.com, with 'officejs.' in white and '.com' in orange.

Unified Cloud Storage
with JIO

Cédric Le Ninivin – cedric.leninivin@nexedi.com

© Nayukeji 2013 – All rights reserved– Confidential

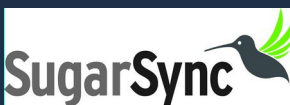
1/ Last month Nayukeji presented officeJS.com, at TechCrunch Shanghai. OfficeJS is a html5 application store. Today I'd like to bring to you a problem HTML5 developer are facing.

Suppose that a developer wants to develop a Javascript application that is compatible with most Cloud providers...

© Nayukeji 2013 – All rights reserved– Confidential

2/ Suppose that a developer wants to develop a Javascript application that is compatible with most Cloud providers...

Cloud Storage APIs



A Developer Nightmare!

© Nayukeji 2013 – All rights reserved– Confidential

3/ First he needs to read the api documentation and implement it into his application (amazon S3). It'll take 10 days to have a working system. Then he will redo the whole process for the next cloud storage (Dropbox), If he wants to use Qiniu 10 more days, etc... If 50 cloud storage provider -> 500 days. This is what we call a developer nightmare.

Suppose that the same developer wants to be compatible with most Content Management System...

© Nayukeji 2013 – All rights reserved– Confidential

4/ Suppose that the same developer wants to be compatible with most Content Management System...

CMS Storage APIs



A Developer Nightmare!

© Nayukeji 2013 – All rights reserved– Confidential

5/ He will proceed the same way. First he will read documentation and implement it (drupal). 10 days. He can use CMIS, but it is not widely spread. 10 more day.... In the end the developer will spend 2 year only on interfaces and not on his application. All These API are still under active development, so every year this developer will need to spend half a year to update his interfaces. This is a real developer nightmare because it prevents interoperability at reasonable cost.

....

Now, before going any further I will set-up my demo. Here in my hand I have a RikoMagic Server running a full ERP system for small to medium company. In my other hand I have a company router I will use for wifi access to the server. Now, I will plug them with USB into my computer for power supply. First the router. Then, the server.

....

Ok, now we can continue. So we were facing a problem, having a html5 application working with most cloud storage provider and CMS.

Suppose now that all this is possible by learning only one simple API...

© Nayukeji 2013 – All rights reserved– Confidential

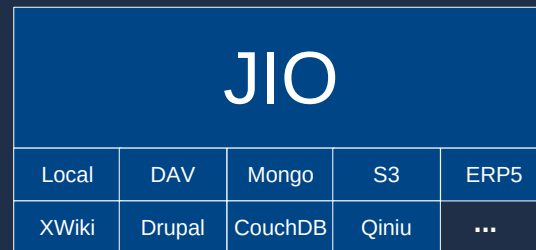
6/ Suppose now that all this is possible by learning only one simple API...

JIO: One API for All Clouds

- jio.get
- jio.put
- jio.post
- jio.remove
- jio.allDocs

Based on CouchDB API
with extensions for queries

One API ↓ get, put, post, remove, allDocs



Any API

© Nayukeji 2013 – All rights reserved– Confidential

7/ JIO

One API for all clouds. Based on CouchDB API, it works with 5 simple methods.

- jio.get: to get document
- jio.put: to update document
- jio.post: to create document
- jio.remove: to remove document
- jio.allDocs: to query the document list.

So How does it work. The html5 application use this 5 methods to work with Jio. And Jio use specific connector to communicate with storages.

JIO: Cloud, better than Cloud

- Encryption
- Splitting
- Indexing
- Online
- Offline
- Replication

Enterprise Grade
=
Trade Secret Protection
+
Performance Optimization

© Nayukeji 2013 – All rights reserved– Confidential

8/ Why do we need Jio: To offer cloud better than cloud.

With jIO we want to offer enterprise grade quality:

- Indexing: fast query
 - Splitting
 - Encryption: security
 - Online and Offline work
 - Replication
- > Trade secret + better performance

jIO Docs

<http://jio.readthedocs.org/en/latest/>

A screenshot of the jIO documentation page. The page has a dark blue header with the title 'jIO Docs' and the URL 'http://jio.readthedocs.org/en/latest/'. Below the header, the main content area is white. On the left side, there is a sidebar with the jIO logo (a stylized 'J' and 'O' with a power symbol), a 'Star' button showing 10 stars, and links for 'Getting Started', 'Downloads', 'Support', 'Stay Informed', and 'Quick search'. The main content area has a 'Welcome to jIO' section, followed by a description of jIO as a JavaScript library for managing JSON documents. It then has sections for 'How does it work?' and 'Copyright and license'. A 'Fork me on GitHub' button is in the top right corner, and a 'v: latest' dropdown menu is in the bottom right corner.

Welcome to jIO

jIO is a JavaScript library that allows to manage JSON documents on local or remote storages in asynchronous fashion. jIO is an abstracted API mapped after CouchDB, that offers connectors to multiple storages, special handlers to enhance functionality (replication, revisions, indexing) and a query module to retrieve documents and specific information across storage trees.

How does it work?

jIO is composed of two parts - jIO core and storage libraries. The core makes use of storage libraries (connectors) to interact with the associated remote storage servers. Some queries can be used on top of the `.allDocs()` method to query documents based on defined criteria.

jIO uses a job management system, so each method call adds a job into a queue. The queue is copied in the browser's local storage (by default), so it can be restored in case of browser crash. Jobs are invoked asynchronously and ongoing jobs are not able to re-trigger to prevent conflicts.

Copyright and license

jIO is open source and is licensed under the [LGPL](#) license.

© Nayukeji 2013 – All rights reserved– Confidential

9/ I won't detail the api, but we have documentations, I'll give the link again at the end of a demo.



10/ I'll show you how we use it. For the demo, I'll start with a simple Task Manager. We presented it at TechCrunch. The task manager itself is not very interesting, what is interesting is the design.

Based on standard HTML5

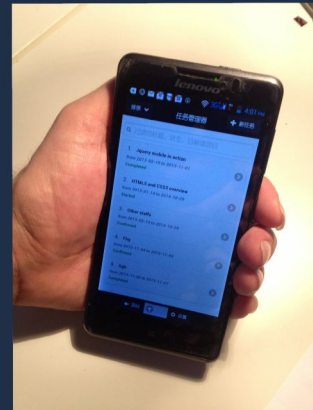
Use Jio for storage/CMS access -> easy storage access

Use JQuery Mobile for rendering - > works on all platforms, all browsers

JIO-Local Storage on Android/iOS

Wifi SSID: officejs

Web: <http://10.0.0.200:12001/tc>



© Nayukeji 2013 – All rights reserved– Confidential

11/ Now to access it:

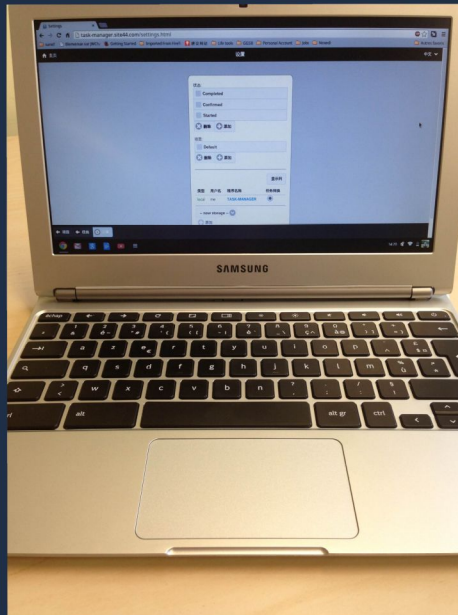
have iphone, android, desktop? Access wifi parameters, change to officeJS (the wifi of the company router)

Please go to a recent browser (I do the same on laptop) and access url: 10.0.0.200:12001/tc

Do you have it?

Add a task. See that the task appear. Close tab and reopen (you can even reboot phone). Task still here. It is using local storage.

JIO-Qiniu on Desktop



© Nayukeji 2013 – All rights reserved– Confidential

12/ Now what if i want to save it online to share it among my devices/browsers?

As we are here let's use Qiniu. (I have a Qiniu account. See bucket. I go to setting, and add storage acces. No task so far (my bucket is empty)).

I'll add one. 2 files in my Qiniu bucket, one is index, the other one document. Now I add a second task. It appear in task list. I go to Qiniu, new document. I download one document.). It is store as JSON. (I download and open index). Index, referring to files. So thanks to that I can synchronise my application data.

Now let's see how we push data on Qiniu in application.

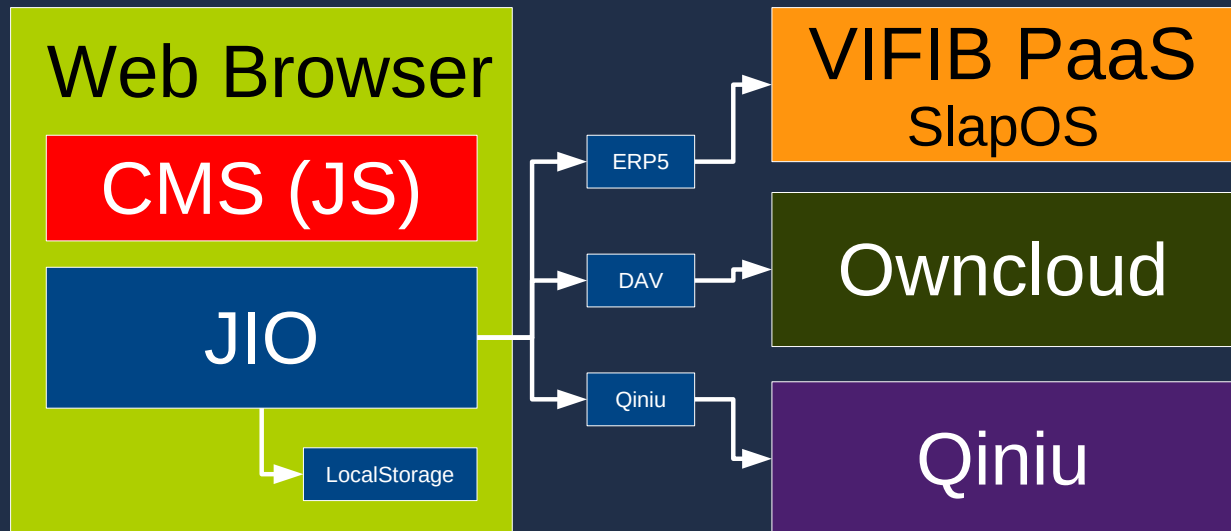
Only One Source Code!

```
jio_project.post(  
  { "type": "Task Report",  
    "project": project,  
    "reference": uuid(),  
    "modified": new Date()  
  },  
  function (err, resp) {  
    // ...  
  }  
);
```

© Nayukeji 2013 – All rights reserved– Confidential

13/ One method. Same code for all storage. No reference to Qiniu nor local. My code is independant of the storage I use.

Demo 2: CMS in JS



© Nayukeji 2013 – All rights reserved– Confidential

14/ So now, let's push forward the idea. Let's build a CMS in html5. Here is the architecture. CMS interact with jIO. jIO store and fetch data from LocalStorage, Qiniu... using its connector.

(I go to browser and show CMS in JS) A Title, a list of document. If I click a document, I get its content. So let's see how it is made. (I show source code). Small html5. I import jIO libraries, have a tilte, two elements (one for index, one for content), and I call a script. Let's see the script.

Here is how I configure my jIO storage using local storage (I previously pushed document in it, so no function to push it). Here I create (not used). Here I fetch documents, Here I build index, Here I display content. Now I'd like to show you how it works in live. (I go back to website, open developer console. I copy configuration. create storage, push document, reload, here it is). Now I'll do the same with Qiniu, for that I need to modify the JS code. My website is host in SlapRunner PaaS, the same we run on rikomagic server and it is being deployed at Hangzhou Dianzi University. Here it is. In it I can monitor process, deploy new component, etc... Here I'll use the file manager to acces and modify my JS. (I open the JS file). I have a confession to make, I do not know how Qiniu API works, but I

Conclusion

- JIO
 - One simple API
 - All clouds
 - All countries
- Offline HTML5
 - Scalability at zero
 - All PC
 - All phones

¥10,000,000 saved by your startup

[http://jio.readthedocs.org/en/latest/
officejs.com/dev](http://jio.readthedocs.org/en/latest/officejs.com/dev)

© Nayukeji 2013 – All rights reserved– Confidential

15/ To conclude.

We saw two things today:

jIO: One Simple API for all clouds in all countries (If I'm here I use Qiniu, in USA dropbox, in europe OVH)

- I do HTML5 application: I only host static file in a small server and use a CDN service (we provide of 0). With this I have scalability at 0 cost and because HTML5 is Standard it works on all PC, all phone -> all platforms, all screen sizes.

With application using jIO and HTML5 I can offer flexibility and scalability at 0 cost and save 10.000.000 RnB for my startup.

So If you want to learn more about jIO you have the documentation here and to know more about HTML5 application development you can look here.

jIO is currently under active development, it use LGPL License and we are looking for contributors and feedback.

Thank you for your attention