

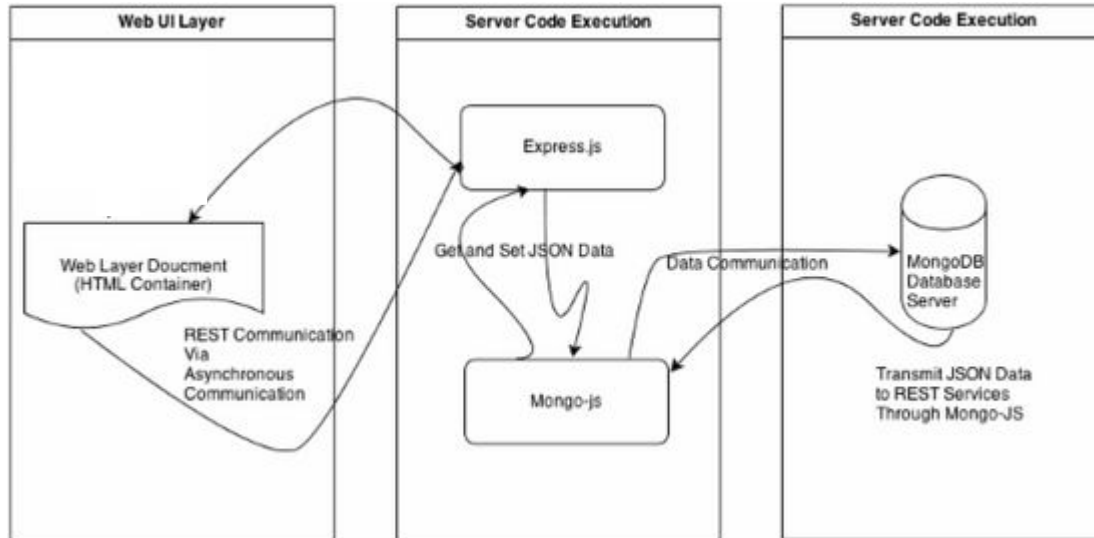
Metadata indexing/retrieval in Databases



**Web application with Nodejs and
MongoDB**

System Architecture Review and Project Structure

- Server: Nodejs
- Client: JS, HTML/Jade
- DB: MongoDB (No-SQL, JSON store)



```
► node_modules
► public
▼ routes
  gallery.js
  index.js
  model_user.js
  random.js
  users.js
▼ views
  ► css
  ► desigual
  ► font-awesome
  ► images
  ► images_insta
  ► js
    error.jade
    gallery.jade
    index.jade
    layout.jade
    random.jade
  README.md
  app.js
  npm-debug.log
  package.json
```

DATA INTEGRATION: PENTAHO ETL

The screenshot displays the Pentaho Spoon interface for configuring an ETL job. The main canvas shows a workflow with two steps: 'JSON Input 3 2' and 'MongoDB Output 2'. Two configuration windows are open in the foreground.

JSON Input Configuration (Left Window):

- Step name: JSON Input 3 2
- Source from field: ☐ Source is from a previous step: ☐ Select field: all
- Use field as file names: ☐ Read source as URL: ☐ Do not pass field downstream: ☐
- File or directory: Add Browse
- Regular Expression:
- Exclude Regular Expression:
- Selected files:
 - File/Directory
 - 1 D:\insta\NodeJS_Photo_Metadata-master\views\ir
 - 2 D:\insta\NodeJS_Photo_Metadata-master\views\ir
 - 3 D:\insta\NodeJS_Photo_Metadata-master\views\ir Delete Edit
- Show filename(s):

MongoDB Output Configuration (Right Window):

- Step name: MongoDB Output 2
- Configure connection | Output options | **Mongo document fields** | Create/drop indexes

Name	Mongo document path	Use field name	NULL values	JSON	M
all	informacio	N	Ignore	Y	N
(filename)	filename	N	Ignore	N	N
(image_filename)	image_filename	N	Ignore	N	N

- Get fields Preview document structure
- Help OK Cancel

METADATA STRUCTURE: Images

InstaFotos 0.002 sec.		
Key	Value	Type
> (42) ObjectId("58c46535ae0e4b04a486dea9")	{ 15 fields }	Object
▼ (43) ObjectId("58c46535ae0e4b04a486deaa")	{ 15 fields }	Object
_id	ObjectId("58c46535ae0e4b04a486deaa")	ObjectId
created_time	1430250718	String
▼ images	{ 3 fields }	Object
> thumbnail	{ 3 fields }	Object
> low_resolution	{ 3 fields }	Object
▼ standard_resolution	{ 3 fields }	Object
width	640	Int32
url	https://scontent.cdninstagram.com/hphotos-xf1/t51.2885-15/e1...	String
height	640	Int32
▼ comments	{ 2 fields }	Object
▼ data	[1 element]	Array
▼ [0]	{ 4 fields }	Object
created_time	1430251477	String
▼ from	{ 4 fields }	Object
full_name	Matilda	String
profile_picture	https://igcdn-photos-a-a.akamaihd.net/hphotos-ak-xf1/t51.288...	String
id	23647442	String
username	matilda_beltran	String
text	♥♥♥	String
id	973342399392709703	String
count	1	Int32
> users_in_photo	[0 elements]	Array
link	https://instagram.com/p/2B-_ejoP4F/	String

MongoDB: Updating filename from “.txt” file to “.jpg” file:

admin localhost:27017 INSTADB

```
var cursor = db.InstaFotos.find();
while (cursor.hasNext()) {
  var x = cursor.next();
  print("Before: "+x['filename']);
  x['filename'] = x['filename'].replace('.txt', '.jpg');
  print("After: "+x['filename']);
  db.InstaFotos.update({_id : x._id}, x);
}
```

0.391 sec.

```
Before: D:\insta\NodeJS_Photo_Metadata-master\views\instagram_desigual\2015_04_29_00\10865116_716555311824601_1840891508_n.txt
After: D:\insta\NodeJS_Photo_Metadata-master\views\instagram_desigual\2015_04_29_00\10865116_716555311824601_1840891508_n.jpg
Updated 1 existing record(s) in 36ms
Before: D:\insta\NodeJS_Photo_Metadata-master\views\instagram_desigual\2015_04_29_00\11137873_1643924289173669_1520910361_n.txt
After: D:\insta\NodeJS_Photo_Metadata-master\views\instagram_desigual\2015_04_29_00\11137873_1643924289173669_1520910361_n.jpg
Updated 1 existing record(s) in 1ms
Before: D:\insta\NodeJS_Photo_Metadata-master\views\instagram_desigual\2015_04_29_00\11176027_1382103798786630_1253543576_n.txt
After: D:\insta\NodeJS_Photo_Metadata-master\views\instagram_desigual\2015_04_29_00\11176027_1382103798786630_1253543576_n.jpg
Updated 1 existing record(s) in 1ms
Before: D:\insta\NodeJS_Photo_Metadata-master\views\instagram_desigual\2015_04_29_00\11184675_417554675093228_1526443974_n.txt
After: D:\insta\NodeJS_Photo_Metadata-master\views\instagram_desigual\2015_04_29_00\11184675_417554675093228_1526443974_n.jpg
Updated 1 existing record(s) in 3ms
```

FRONT-END(I): JADE

- HTML Template, based on indentations instead of tags.
- Faster code writing.
- Dynamic content (- each publication in list)

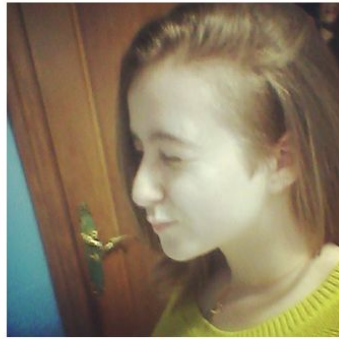
```
#content.gallery-images-container
  .box-gutter
  .box-sizer
  each publication in list
    .box
      a(href=publication.photo)
        img(alt='Agra picture', src=publication.photo)
      center
        p
          | #{publication.user}
        p #{publication.text}]
```



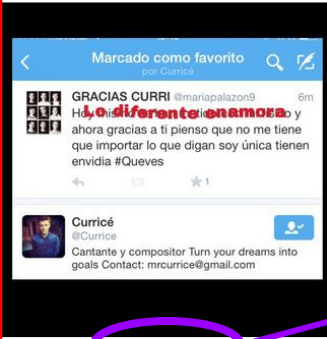
```
<div id="content" class="gallery-image
  <div class="box-gutter"></div>
  <div class="box-sizer"></div>
  <div class="box">
    <a href="desigual/2015_04_28_22/
    desigual/2015_04_28_22/11191367_
    <center>
      <p>eva_sanchez2000</p>
      <p>"#QueVes"</p>
      <p></p>
    </center>
  </div>
  <div class="box">
    <a href="desigual/2015_04_28_22/
    desigual/2015_04_28_22/11191416_
    <center>
      <p>adriana_sombrerita</p>
      <p>"#QueVes"</p>
      <p></p>
    </center>
  </div>
  <div class="box">
    <a href="desigual/2015_04_28_22/
    desigual/2015_04_28_22/11192569_
    <center>
      <p>almudena_vidal</p>
      <p>"Aquello que une y separa
      #ganasdeestarenchantada #lavi
      <p></p>
    </center>
  </div>
  <div class="box">
    <a href="desigual/2015_04_28_22/
    desigual/2015_04_28_22/11192577_
    <center>
      <p>ofelia_roca_roman</p>
      <p>"#QueVes veo a dos persona
      <p></p>
    </center>
  </div>
</div>
```


FRONT-END(II): View

- Gallery example:



plera_anitaa
"#queves"



mpalazon23

"Muchas gracias @currice! Cada día estoy mas orgullosa de ti! Te quiero #queves #currice #curri #septiembre13 #desigual"



matilda_beltran
laura_mccosh

"Beautiful Winnie on the last campaign for Desigual
♥\n\nRepost @matilda_beltran\n\nReport @desigual with @repostapp... Last work for Desigual with @rafitnoymakeup @laura_mccosh and @winnieharlow #lovenmyjob #queves #winnieharlow #makeupartist #hairstylist #maccosmetics #morecanoil #workinghard #workaholic #desigual"

```
<div id="content" class="gallery-imag
<div class="box-gutter"></div>
<div class="box-sizer"></div>
<div class="box">
  <a href="desigual/2015_04_28_22
desigual/2015_04_28_22/11201665
  <center>
    <p>plera_anitaa</p>
    <p>"#queves"</p>
    <p></p>
  </center>
</div>
<div class="box">
  <a href="desigual/2015_04_28_22
desigual/2015_04_28_22/11201685
  <center>
    <p>mpalazon23</p>
    <p>"Muchas gracias @currice
#desigual"</p>
    <p></p>
  </center>
</div>
<div class="box">
  <a href="desigual/2015_04_28_22
desigual/2015_04_28_22/11201789
  <center>
    <p>laura_mccosh</p>
    <p>"Beautiful Winnie on the
@repostapp... Last work fo
#winnieharlow #makeupartist
    <p></p>
  </center>
</div>
</div>
</div>
```

BACK-END: NodeJS

- Event-oriented single-thread requests processing.
- Handles concurrency by callbacks.

```
user do an action
  |
  v
application start processing action
  ↳ make database request
      ↳ do nothing until request completes
request complete
  ↳ send result to user
```


BACK-END: Node JS and Express

-Receiving params + MongoDB query:

```
router.post('/', function(req, res, next) {  
  //TODO:::::::::::  
  //Aquí, hem de fer la query d'alguna manera  
  //Rebem els params del form  
  var usr= req.body.username  
  var tags= req.body.tag
```

```
var query2 = { "informacio.caption.text": new RegExp('' + tags) };  
db.collection('InstaFotos').find(  
  { $or: [{"informacio.user.username": usr}, query2]} ,  
  {filename:1, "informacio.user.full_name":1,  
   "informacio.user.username":1,  
   "informacio.caption.text":1}).toArray(function(err, filename){
```

BACK-END: Node JS and Express (II)

- Image path processing (absolute path to relative path):

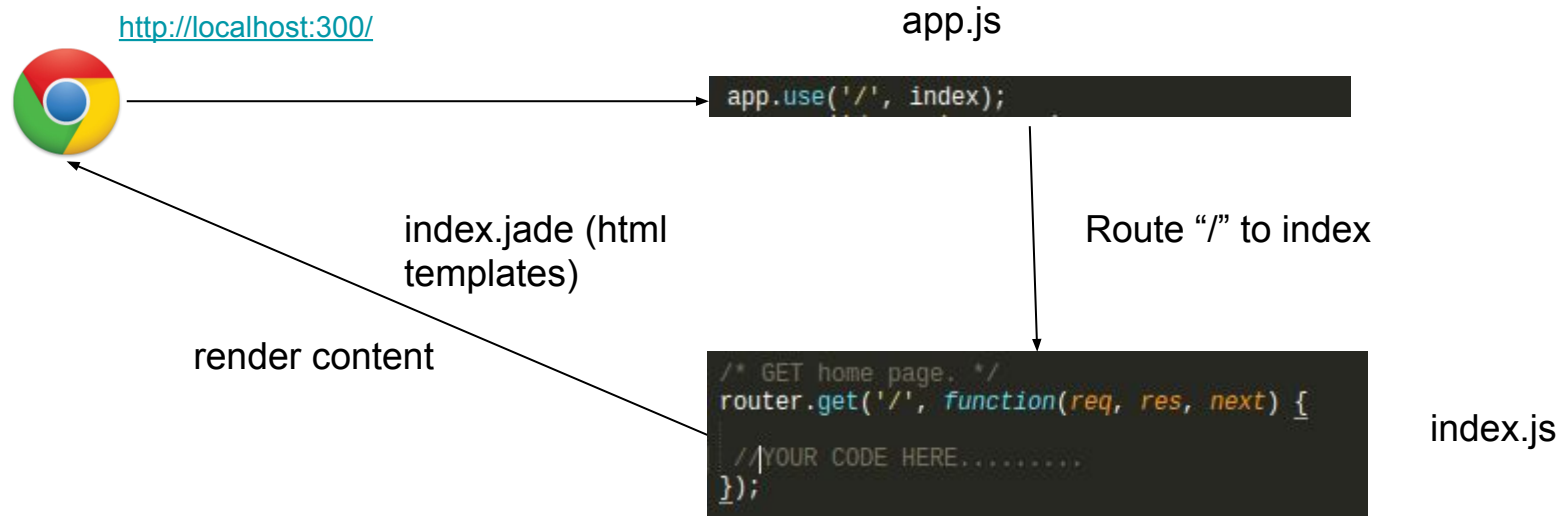
```
for (i in filename){  
    str_res_path[i]=JSON.stringify(filename[i].filename);  
    str_res_path[i]=str_res_path[i].substring(47,str_res_path[i].lastIndexOf(".jpg")+4);  
    str_res_path[i] = str_res_path[i].replace(/\\/g, "/");  
}
```

```
path:  
"D:\\insta\\NodeJS_Photo_Metadata-master\\views\\instagram_desigual\\2015_04_29_19\\11187072_386172014907980_933271521_n.jpg"  
path processat:  
/instagram_desigual/2015_04_29_19/11187072_386172014907980_933271521_n.jpg
```

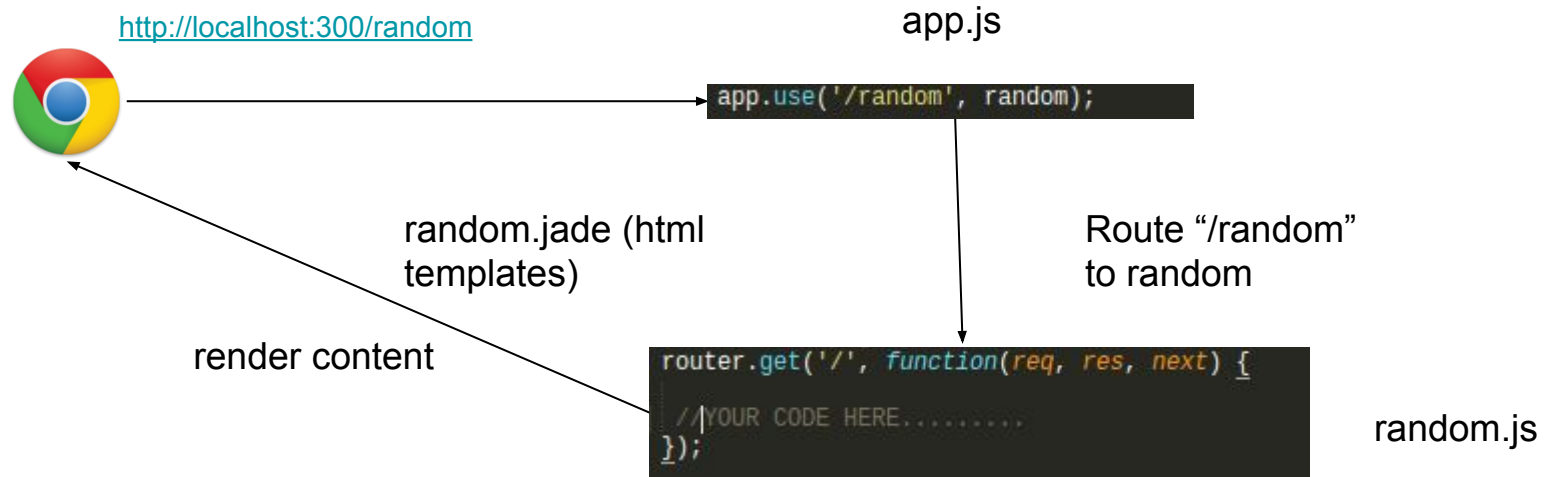
- Returning results:

```
publications.list.push({  
    "photo" : temp_photo,  
    "user" : temp_usr,  
    "text": temp_text //S'ha d'afegit  
});  
}  
console.log(publications)  
res.render('gallery', publications);  
});
```

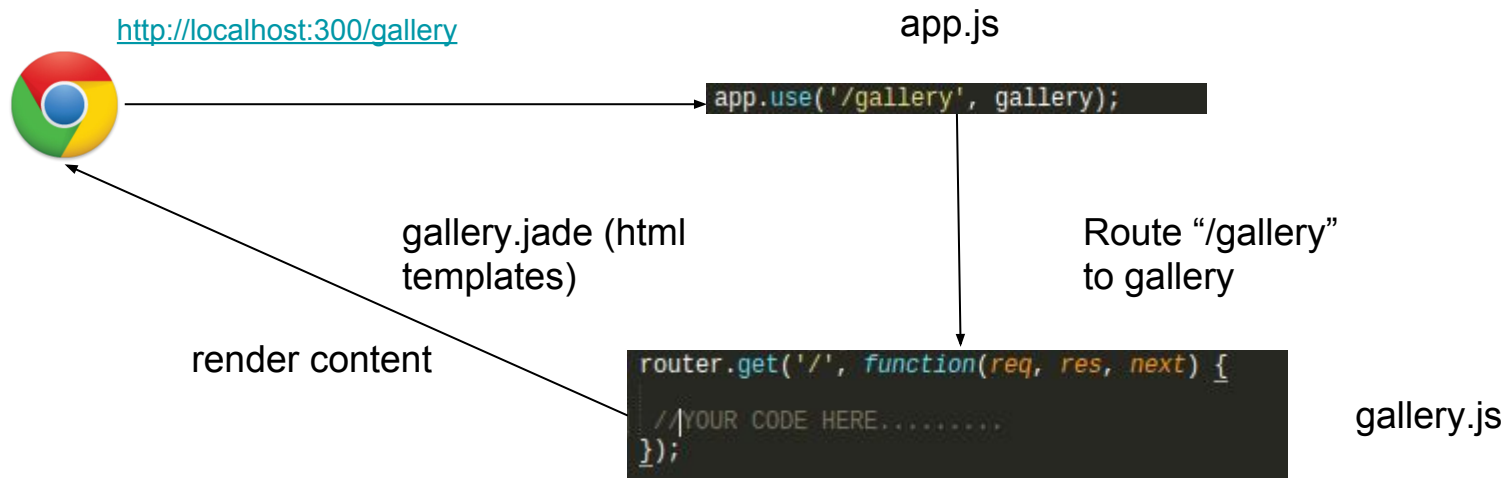
Web App Routes (I): Index



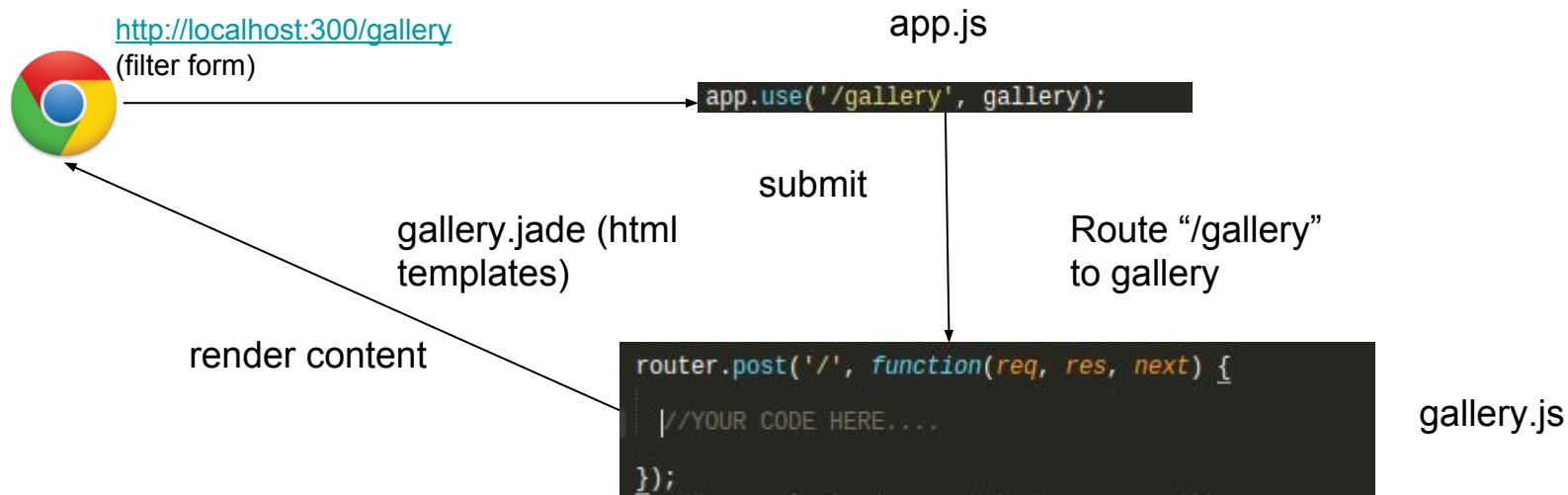
Web App Routes (II): Random



Web App Routes (III): Gallery



Web App Routes (IV): Gallery (form)



Conclusions:

- Using JSON objects both in the database and in the communication between NodeJS and client makes easier the information processing.
- Image's metadata is easier to store on non-relational databases than relational ones.
- We can use the same programming language both on client and server (JavaScript).

Podeu trobar el projecte i la guia per desplegar-lo a:

https://github.com/ericpresas/NodeJS_Photo_Metadata