nuxeo

EXPOSING
A FLEXIBLE, COMPOSABLE & EXTENSIBLE REST API

Thierry Delprat

td@nuxeo.com

https://github.com/tiry/

AGENDA

nuxeo

- Quick introduction
  - *provide some context*
- API design constraints & principles
  - *explain the problem we want to solve*
- Building Nuxeo API
  - *REST + Automation + Composition*
- Design consequences
  - *price of flexibility*

nuxeo

SOME CONTEXT

What we Do and What Problems We Try to Solve

NUXEO

nuxeo

*we provide a **Platform** that **developers** can use to build **highly customized** Content Applications*

*we provide **components**, and the **tools** to assemble them*

*everything we do is **open source***

https://github.com/nuxeo

various **customers** - various **use cases**

Track game builds

Electronic Flight Bags

Central repository for Models

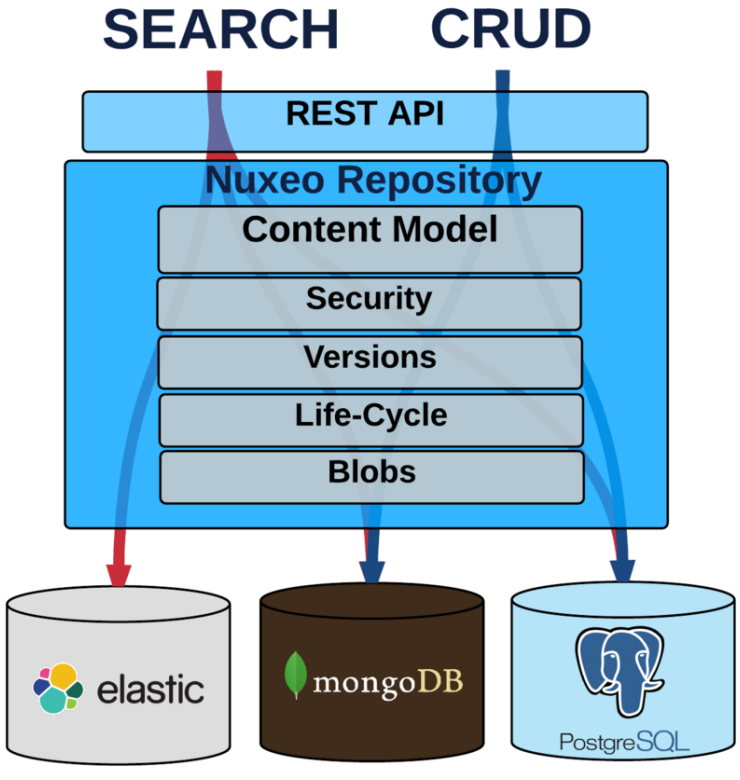
Food industry PLM

me: developer & CTO - joined the Nuxeo project 10+ years ago

NUXEO PLATFORM

nuxeo

Repository

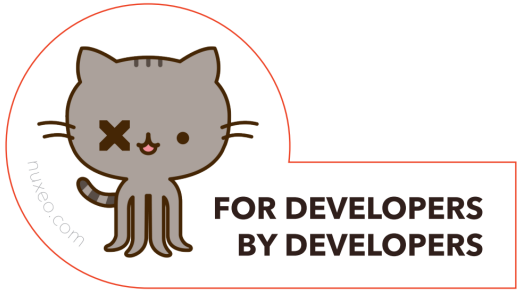


Services

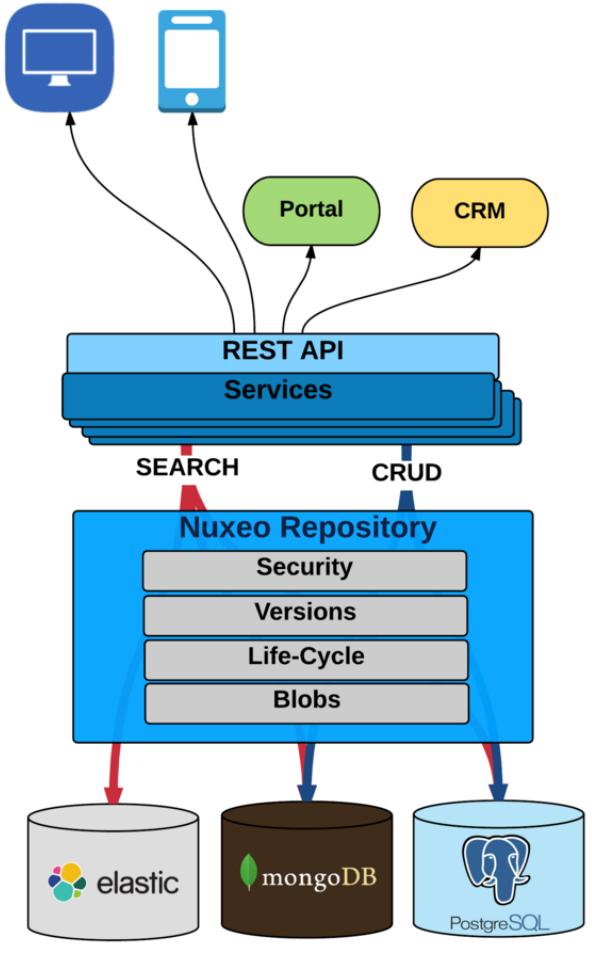
- *Workflows, Conversions, Diff, Notifications, Activity ...*

WHY API IS KEY FOR US

nuxeo

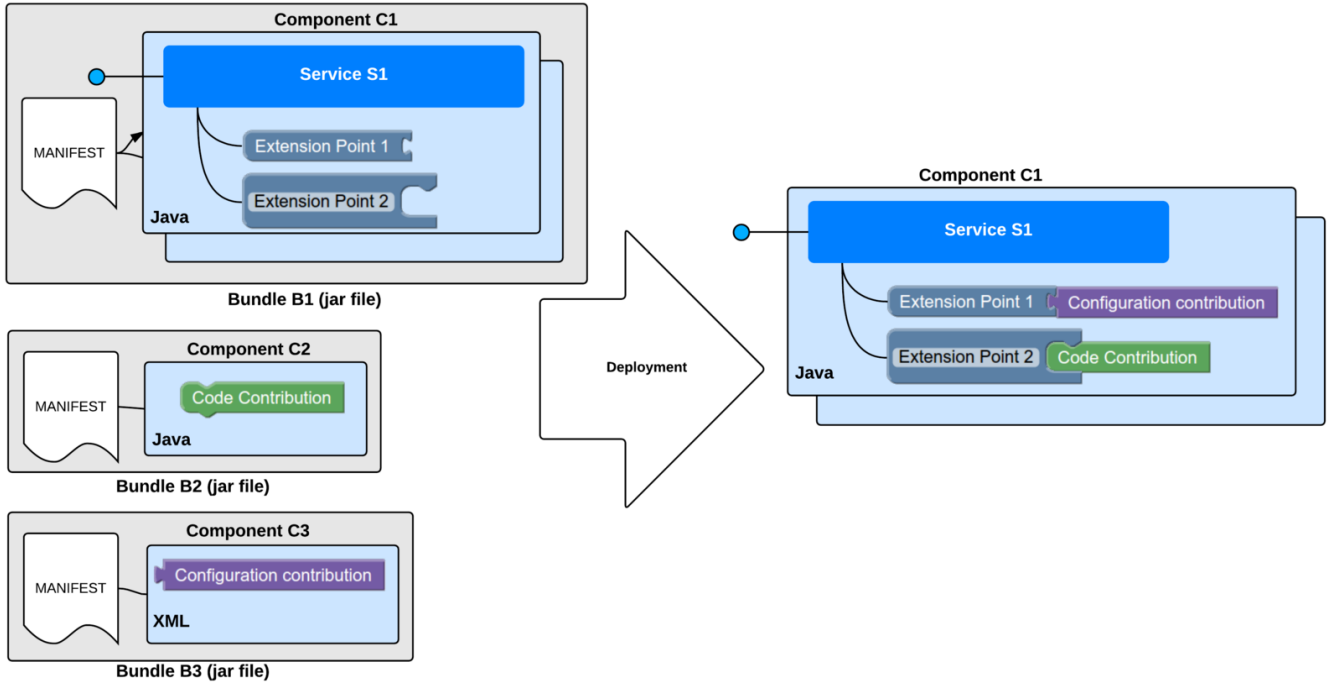


- Nuxeo Repository is a **backend**
  - *Portals, Mobile Apps, ERP, CRM ...*
- API is **UI**
  - *for the developers*
  - *HTML5/JS*



PLUGABILITY CHALLENGE

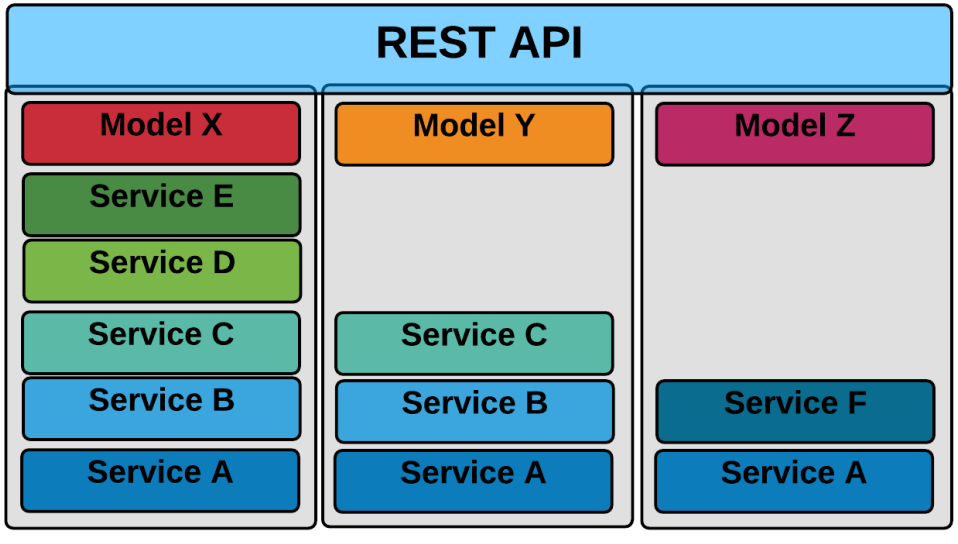
- In Nuxeo architecture **everything is a plugin**
  - *Nuxeo Server can provide a single service or 100's of services*
- Everything is **configurable**
  - *Logic and Data Structrures depends on configuration*



API CHALLENGE

nuxeo

Expose a Platform: not an application



"**One** API"

but

Multiple combinations

of

services, plugins
and Domain Models

developers using the platform
*want to expose the API of **their** Application*

nuxeo

NEED TO FIND A SOLUTION

One Platform ➡ "**One** API"

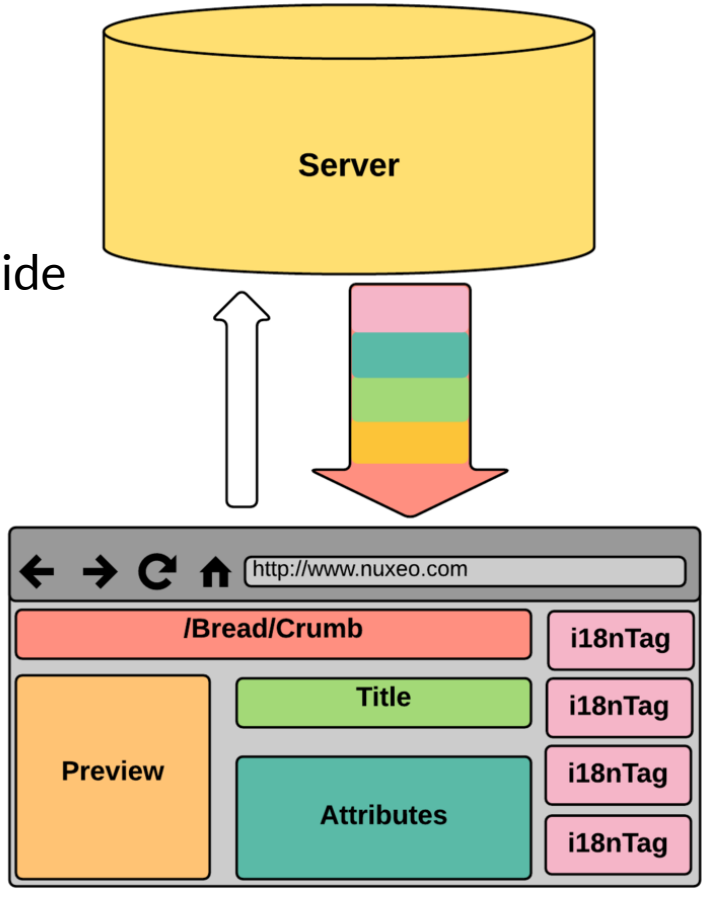
nuxeo

REST API DESIGN PRINCIPLES

what we want to have

BE EFFICIENT

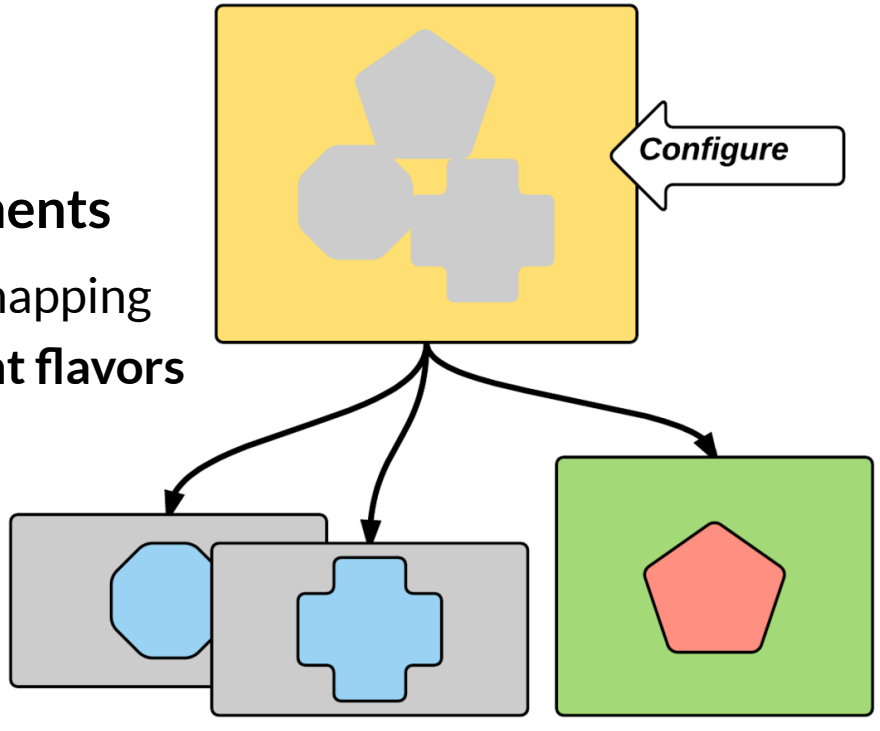
- Avoid **round trips**
  - Get all needed data in one call
  - Resolve some data on the server side
- Avoid fetching **too much data**



BE FLEXIBLE

nuxeo

- **Adapt** to the server side **configuration**
  - Domain model definition
- **Adapt** to client side **requirements**
  - Provide data for the screen mapping
  - Application can have **different flavors**



KEEP IT CLEAN

nuxeo

- Work between **transaction boundaries**
  - do all the work in one call
- Ensure **isolation**
  - Other users should not see inconsistent data
- Maintain **encapsulation**
  - Client should not make assertion on server implementation

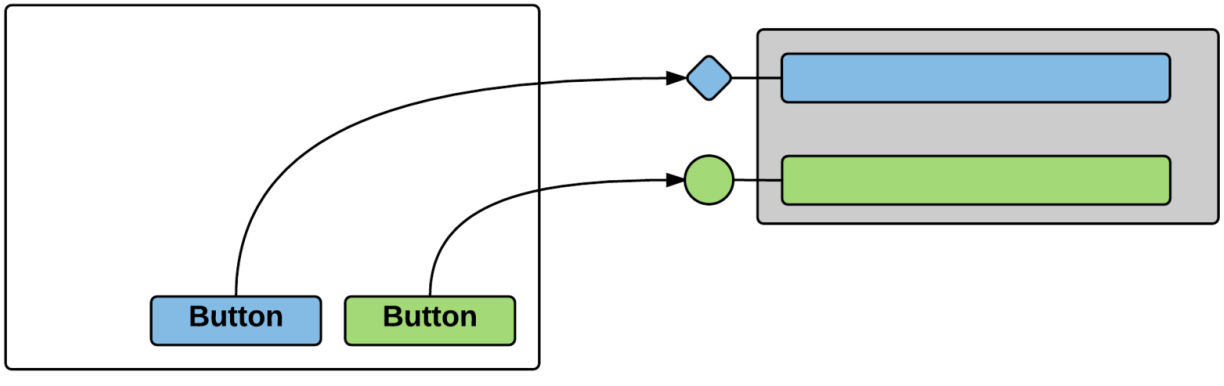
Client consumes a service, it does not build the service.

BE EXTENSIBLE

nuxeo

- Expose ***any*** **meaningful business API**
  - Make API clean and application maintenance easy

- **Adapt** API **granularity** to the target Applications
  - one API behind each single button



We can not build the target Business API: users/devs will do it

BALANCE CLIENT/SERVER ROLES

nuxeo

"one-size-fits-all" does not work

➡ *Client driven*

CLIENTS

Ask for the **data they need**
Use **custom API**

"open bar" seems too messy

➡ *Server controlled*

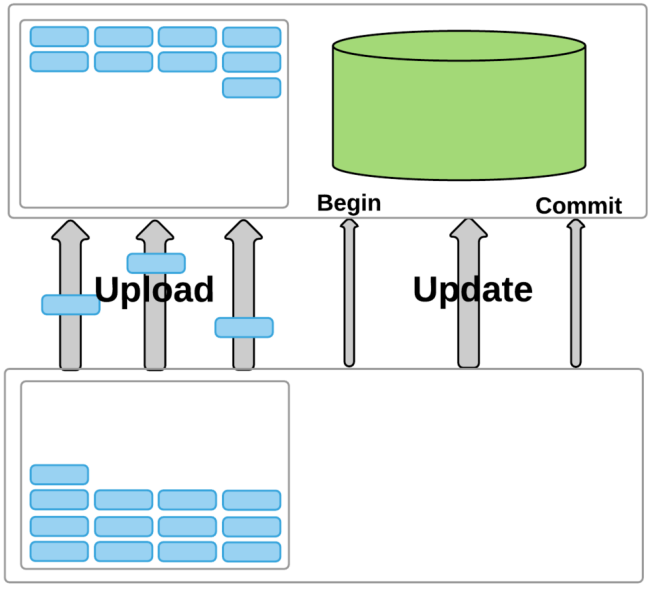
SERVER

Manage the **meta-model**
Choose **what API is exposed**
(versioned software artifact)

BLOB FRIENDLY

- Chunked & Out of band upload

```
Content-Type: multipart/mixed
```

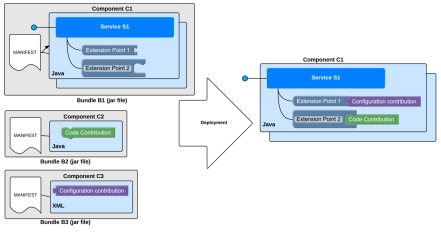


- Cachable and Seekable download

BE SENSIBLE

nuxeo

- Do not **lose our soul**
  - *fight to keep the dynamicity of the platform!*



- No REST integrism
  - *Useful is more important than Beautiful*

- **Dogfooding** is key
  - *if this is not good enough internally, this is not good*

- Building API is **part of the development cycle**
  - *adding http API should never be a task for later*

nuxeo

PRINCIPLES

BUILD SOMETHING THAT WORKS

DISCLAIMER

Actually, just the chronology has been adjusted !

nuxeo

BUILDING THE REST API

Exposing Resources

EXPOSING RESOURCES

nuxeo

Expose **Use Cases** !?

PLATFORM  *Target use cases are not defined*

Expose the **Domain Model** !?

CONFIGURABLE  *Target Domain Model is unknown*

Expose raw technical resources !

EXPOSE SIMPLE RESOURCES

nuxeo

Expose raw resources as EndPoint with REST Bindings

Documents

```
GET /repo/{repoId}/path/{docPath} HTTP 1.1
```

```
GET /repo/{repoId}/id/{docId} HTTP 1.1
```

Users & Groups

```
GET /user/{userName} HTTP 1.1
```

```
GET /group/{groupName} HTTP 1.1
```

```
GET /directory/{directoryName}/{entryId} HTTP 1.1
```

Tasks & Workflows

```
GET /workflowModel/{modelName} HTTP 1.1
```

```
GET /workflow/{workflowInstanceId} HTTP 1.1
```

```
GET /task/{taskId} HTTP 1.1
```

EXPOSE SIMPLE RESOURCES

nuxeo

path : Access documents by their path

Show/Hide | List Operations | Expand Operations | Raw

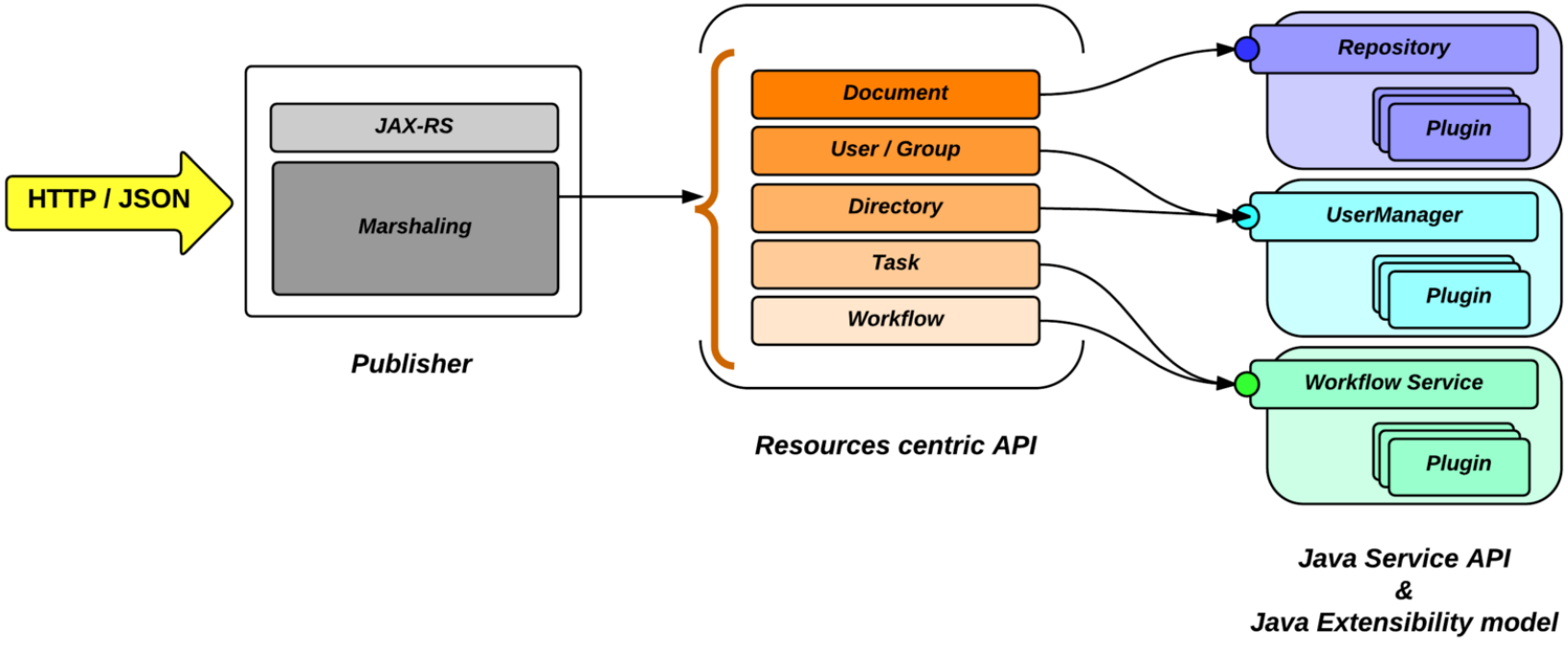
GET	/path/{docPath}	Find a document by its path
PUT	/path/{docPath}	Updates a document by its path
DELETE	/path/{docPath}	Deletes a document by its path
POST	/path/{docPath}	Creates a document by its parent path
GET	/repo/{repoId}/path/{docPath}	Find a document by its path
PUT	/repo/{repoId}/path/{docPath}	Updates a document by its path
DELETE	/repo/{repoId}/path/{docPath}	Deletes a document by its path
POST	/repo/{repoId}/path/{docPath}	Creates a document by its parent path

id : Access documents by their id

Show/Hide | List Operations | Expand Operations | Raw

EXPOSE SIMPLE RESOURCES

nuxeo

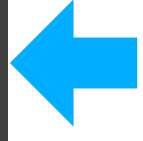


GET A DOCUMENT

nuxeo

```
GET /nuxeo/api/v1/path/movies/star-wars HTTP/1.1
```

```
{
  "entity-type": "document",
  "repository": "default",
  "uid": "5b352650-e49e-48cf-a4e3-bf97b518e7bf",
  "path": "/movies/star-wars",
  "type": "MovieCollection",
  "isCheckedOut": true,
  "title": "Star Wars",
  "facets": [
    "Folderish"
  ]
}
```

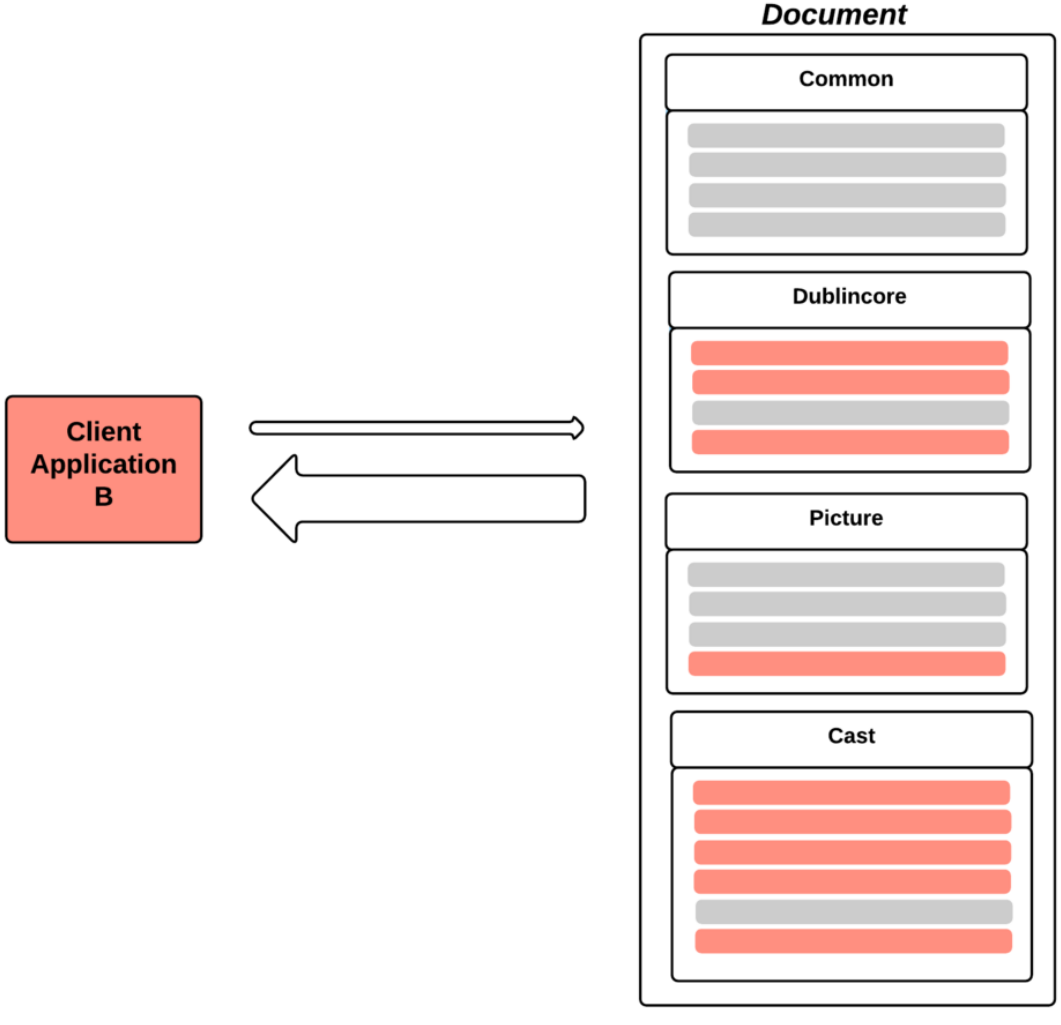


Server returns a minimal payload

ADAPTATIVE MARSHALING

nuxeo

Client need to control what **data schemas** are sent



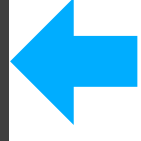
ADAPTATIVE MARSHALING

nuxeo

- Control what **data schemas** are sent to the client

```
GET /nuxeo/api/v1/path/movies/star-wars HTTP/1.1
X-NXProperties dublincore, common
```

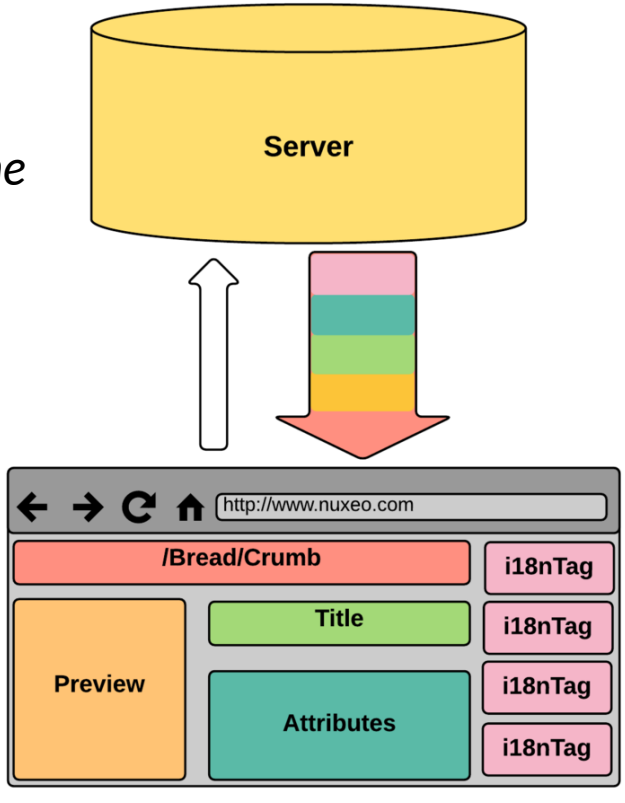
```
{
  "entity-type": "document",
  "repository": "default",
  "uid": "5b352650-e49e-48cf-a4e3-bf97b518e7bf",
  "path": "/movies/star-wars",
  "type": "MovieCollection",
  "isCheckedOut": true,
  "title": "Star Wars",
  "properties": {
    ...
    "common:icon": "/icons/movieCollection.png",
    "dc:description": "Star Wars collection",
    "dc:creator": "tiry",
    "dc:modified": "2015-10-22T02:12:59.07Z",
    "dc:lastContributor": "tiry",
    "dc:created": "2015-10-22T02:12:59.07Z",
    "dc:title": "Star Wars",
    ...
    "dc:contributors": [tiry, "system" ]
  },
  "facets": [
    "Folderish"
  ]
}
```



FETCHING CONTEXTUAL DATA

nuxeo

- Client may require **more data**
  - *get Document **children** at the same time*
  - *get the **breadcrumb** data*
  - *get **thumbnail** or **preview** url*
  - ...
- Client **ask for the data**
  - *using **Headers***
  - *using Query String **parameters***

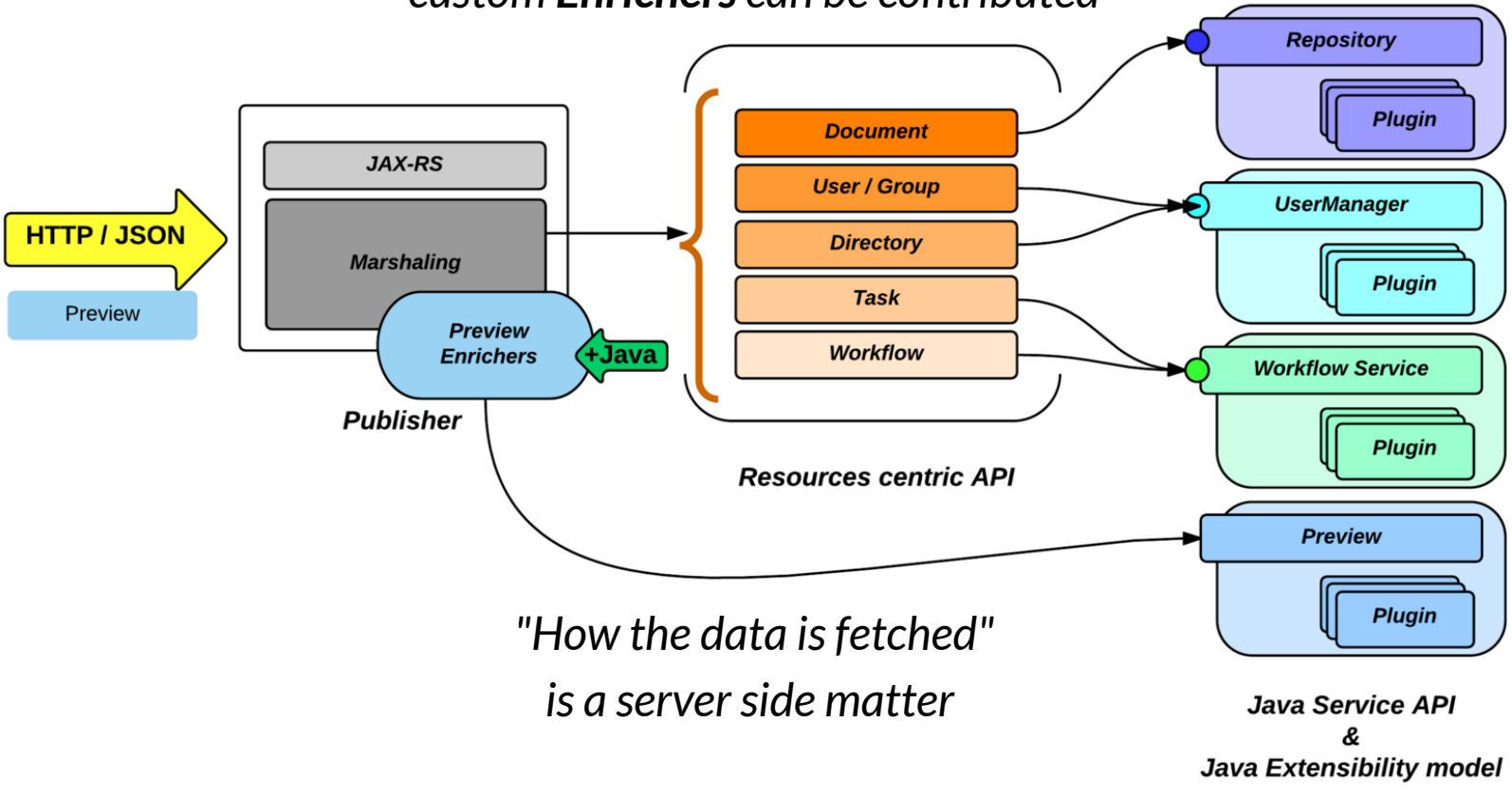


FETCHING CONTEXTUAL DATA

nuxeo

Marshaling registry is pluggable

*custom **Enrichers** can be contributed*



"How the data is fetched"

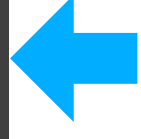
is a server side matter

FETCHING CONTEXTUAL DATA

nuxeo

```
GET /nuxeo/api/v1/path/movies/star-wars?enrichers.document=thumbnail H
```

```
GET /nuxeo/api/v1/path/movies/star-wars HTTP/1.1
X-NXenrichers.document: thumbnail
```



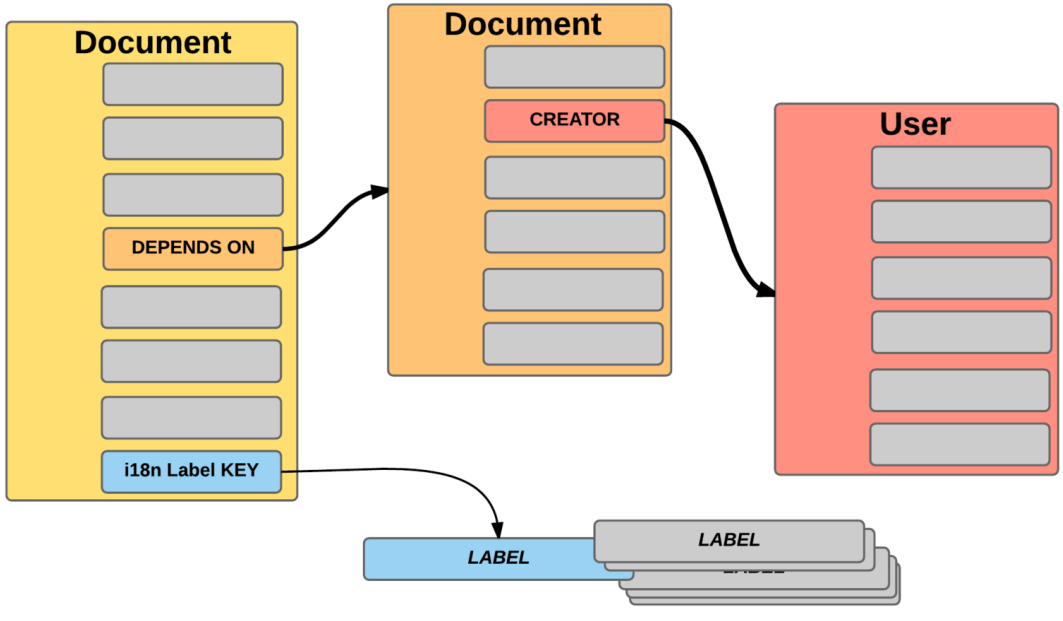
```
{
  "entity-type": "document",
  "repository": "default",
  "uid": "5b352650-e49e-48cf-a4e3-bf97b518e7bf",
  "path": "/movies/star-wars",
  "type": "MovieCollection",
  "isCheckedOut": true,
  "title": "Star Wars",
  "contextParameters":
   {
     "thumbnail":
     {
       "url": "/nuxeo/nxthumb/default/5b352650-e49e-48cf-a4e3-bf97b518
     }
  },
  "facets": [
    "Folderish"
  ]
}
```

RETRIEVE LINKED DATA

nuxeo

- **Resolve** entity fields
  - *pointing to a label*
  - *pointing to an other Document*
  - *pointing to a User*
  - ...

`Implicit JOIN`



RETRIEVE LINKED DATA

nuxeo

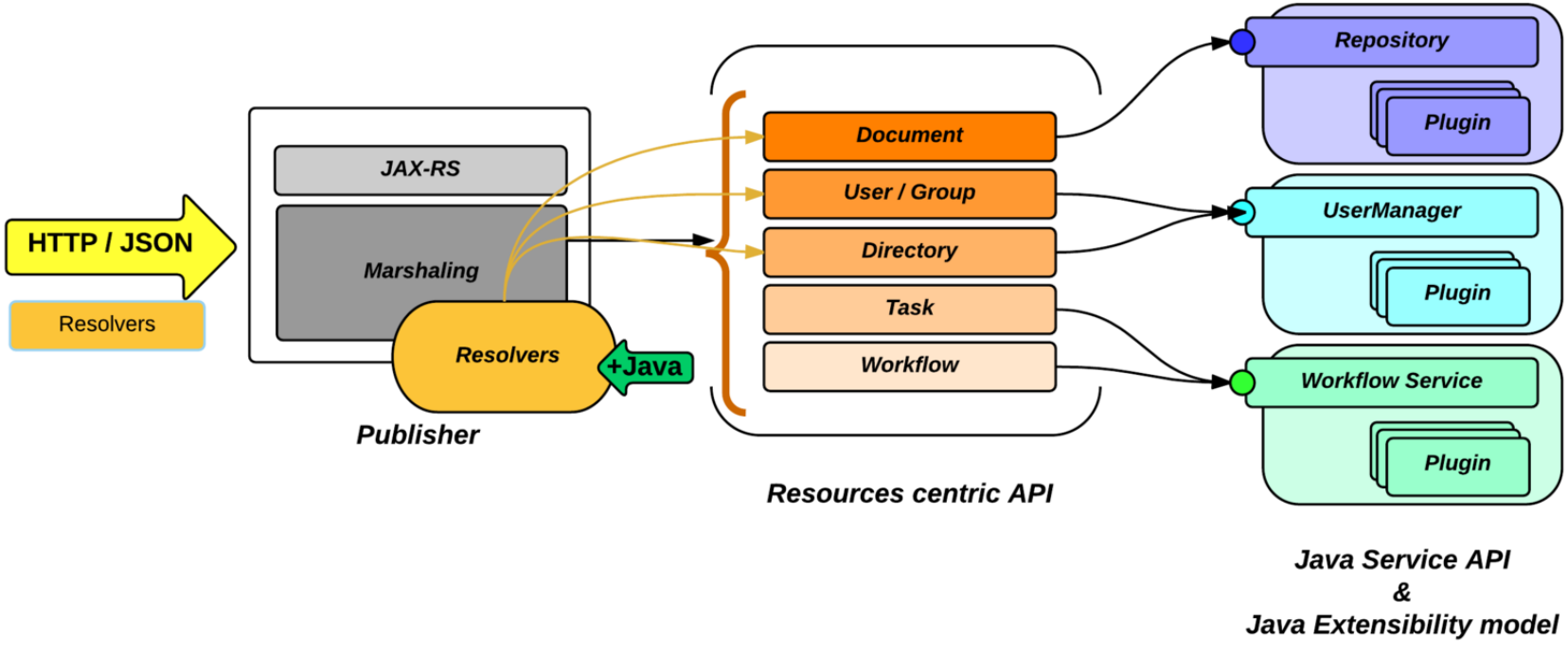
- Use client side parameter to know what to resolve
  - ***header***
  - *QueryString* ***parameter***

```
fetch.objectType=fieldToFetch

translate.objectType=fieldToTranslate

depth=children
```

- Can be **recursive**
  - *client need to control that too!*

RETRIEVE LINKED DATA

nuxeo



ADAPTERS

nuxeo

- Change the **return type**
  - *get only **ACLs** or **History** info about the Document*
  - *get the **tasks** associated to document*

```
GET /nuxeo/api/v1/path/movies/star-wars@acl HTTP/1.1
```

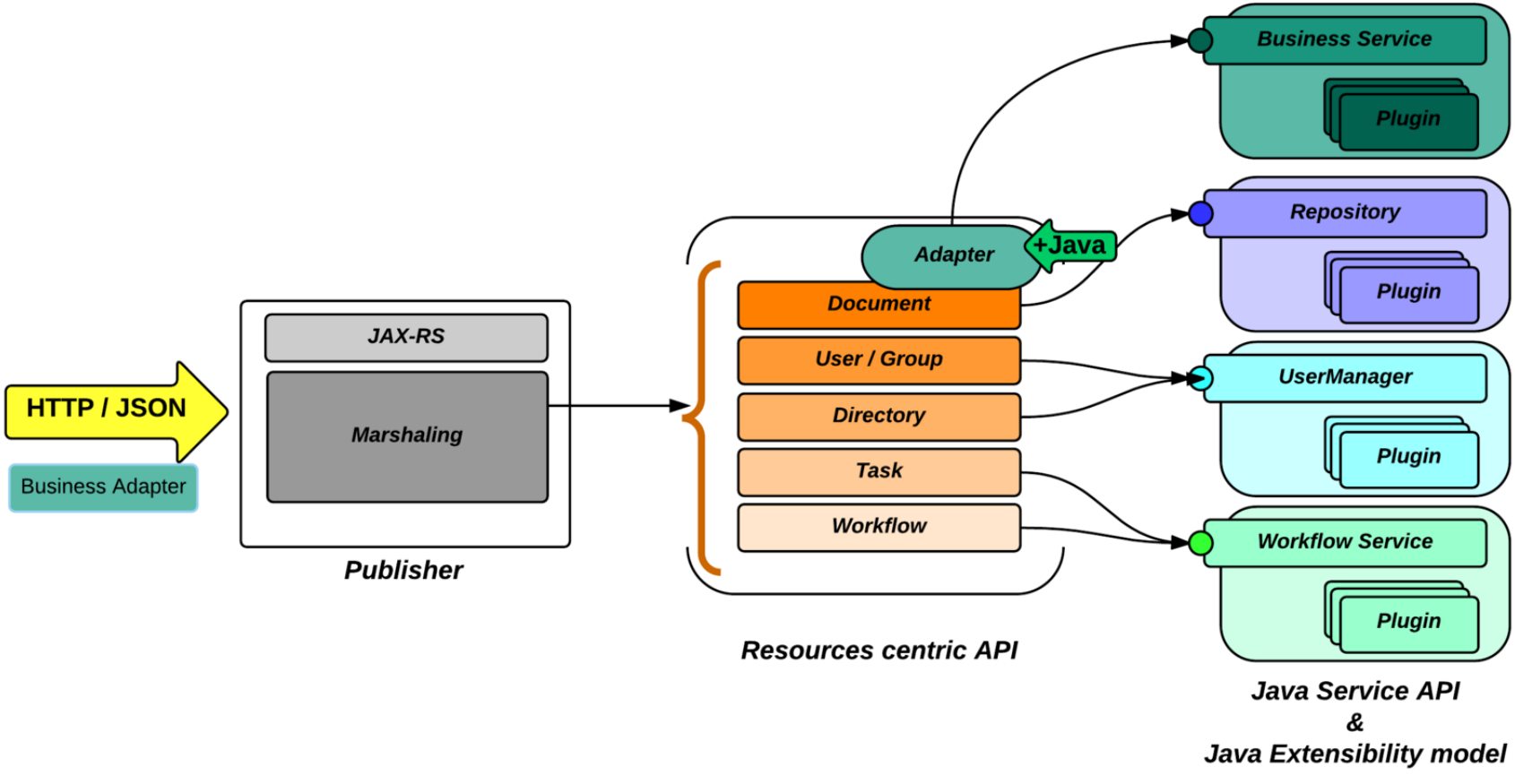
```
GET /nuxeo/api/v1/path/movies/star-wars@audit HTTP/1.1
```

- Use your own **business object**
  - use **business Adapters**
    - wrap document or documents
    - provide custom marshaling

```
GET /nuxeo/api/v1/path/movies/star-wars@bo/MyBusinessObject HTTP/1.1
```

ADAPTERS

nuxeo

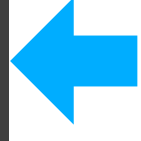


ADAPTERS

nuxeo

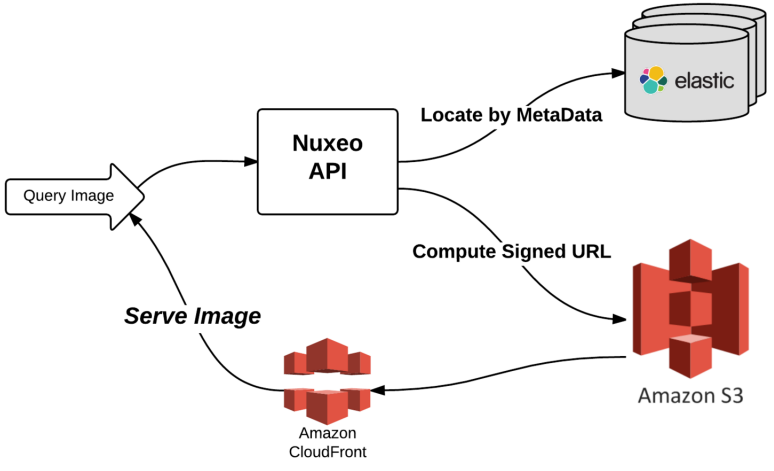
```
GET /nuxeo/api/v1/path/movies/star-wars@bo/MovieCollection HTTP/1.1
```

```
{
    entity-type: "MovieCollection"
    id: "5b352650-e49e-48cf-a4e3-bf97b518e7bf",
    "title": "Star Wars"
    "episodes": 7
}
```

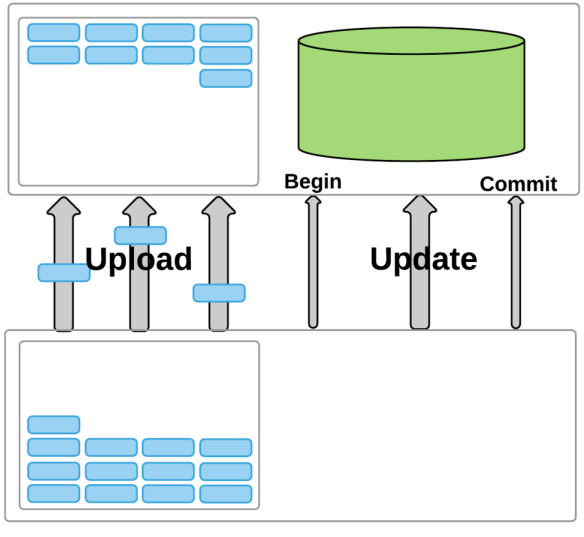


BLOBS

- **Sent as links**
  - Digest
  - CDN



- **Uploaded out-of-band**
  - chunking
  - reference in JSON



BLOB UPLOAD

nuxeo

- **Upload EndPoint**

```
POST /api/v1/upload/{batchId}/{fileIdx} HTTP 1.1
X-Upload-Chunk-Index 0
X-Upload-Chunk-Count 5
```

- **Reference** Blobs from JSON Payload

```
PUT /nuxeo/api/v1/path/movies/star-wars HTTP/1.1
```

```
{"entity-type": "document",
 "properties": {
  {
   "file:content" : {
     "upload-batch' : "0b0061d48f69b072",
     "upload-fileId" : 0,
     "type" : "blob"
    }
}}
```

ARE WE HAPPY WITH THAT ?

nuxeo

- **Efficiency**
  - *we can get all data in one call*

- **Flexibility**
  - *we can configure the data we want*

- **Extensibility: partial**
  - *enrichers, resolvers & adapters are not always enough*

- **Coverage: poor**
  - *100+ services and only 5 endpoints*
  - *not everything is CRUD*

nuxeo

NEED A WAY TO MAP 100+ SERVICES

Without creating 100 endpoints!

Need an other paradigm !

nuxeo

AUTOMATION API

Exposing service API over HTTP

PRINCIPLES

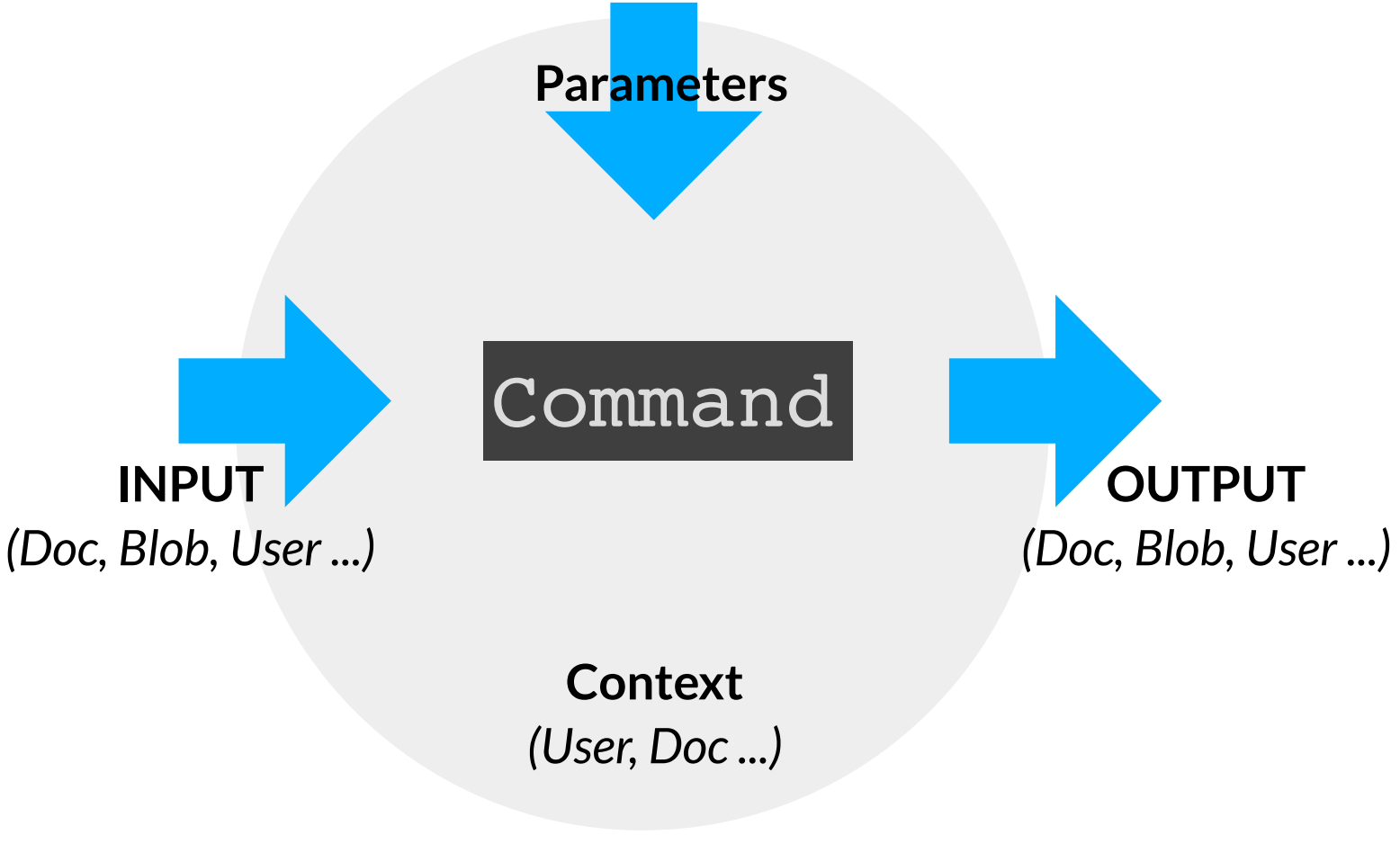
nuxeo

- Build a **coarse gained API** on top of service Java API
  - select simple `Commands`
- **Shell like** commands !
  - *each service can contribute*

```
> commandA(p1,p2) | commandB(p3,p4)
```

COMMAND SYNOPSIS

nuxeo



COMMANDS

nuxeo

WebUI.AddErrorMessage WebUI.AddInfoMessage WebUI.AddMessage Document.AddPermission Document.AddToCollection DocumentMultivaluedProperty.addItem Task.ApplyDocumentMapping Blob.AttachOnDocument BlobHolder.AttachOnCurrentDocument AttachFiles Audit.QueryWithPageProvider Blob.ImportClipboard Blob.ImportWorklist Blob.RunConverter Document.BlockPermissionInheritance WorkflowModel.BulkRestartInstances Business.BusinessCreateOperation Business.BusinessFetchOperation Business.BusinessUpdateOperation Navigation.GoBack Workflow.Cancel Navigation.ChangeCurrentTab Document.CheckIn Document.CheckOut Update.Workflow Workflow.CreateRoutingTask Document.DeleteSelectedDocuments WebUI.ClearWorklist WorkflowTask.Complete Blob.ConcatenatePDFs Context.FetchDocument Context.FetchFile Blob.ToPDF Blob.Convert Document.Copy Document.Create FileManager.Import UserWorkspace.CreateDocumentFromBlob Seam.CreateDocumentInUI Picture.Create Document.CreateLiveProxy Document.AddRelation Collection.Create Workflow.CreateRoutingTask Task.Create Directory.CreateEntries Document.Delete Document.DeleteRelation Directory.DeleteEntries WebUI.DestroySeamContext Repository.GetDocument Document.Export WebUI.DownloadFile Blob.ExportToFS Document.FetchByProperty Blob.CreateFromURL FileManager.ImportInSeam FileManager.ImportWithMetaData FileManager.ImportWithMetaDataInSeam Document.Filter Document.FollowLifecycleTransition Comment.Moderate Document.GetBlobs Document.GetChildren Document.GetChildrenTree Document.GetBlobsByProperty User.GetUserWorkspace Document.GetLinkedDocuments Proxy.GetSourceDocument User.Get Document.GetParent Context.GetEmailsWithPermissionOnDoc Context.GetTaskNames Context.GetUsersGroupIdsWithPermissionOnDoc Document.GetVersions Directory.Projection Collection.Suggestion User.GetCollections Directory.Entries Directory.SuggestEntries Collection.GetDocumentsFromCollection Favorite.GetDocuments Document.Routing.GetGraph Picture.GetView Workflow.GetOpenTasks Tag.Suggestion Task.GetAssigned UserGroup.Suggestion Document.GetRendition Blob.PostToURL Image.Blob.Resize WebUI.InitSeamContext JsonStack.ToggleDisplay Actions.GET GetRepositories Document.Lock Log Audit.LogEvent Auth.LoginAs Auth.Logout Document.Move Document.PublishToSections NRD-AC-PR-ChooseParticipants-Output NRD-AC-PR-LockDocument NRD-AC-PR-UnlockDocument NRD-AC-PR-ValidateNode-Output NRD-AC-PR-force-validate NRD-AC-PR-storeTaskInfo WebUI.NavigateTo NuxeoDrive.SetActiveFactories NuxeoDrive.AddToLocallyEditedCollection NuxeoDrive.AttachBlob NuxeoDrive.CanMove NuxeoDrive.CreateFile NuxeoDrive.CreateFolder NuxeoDrive.CreateTestDocuments NuxeoDrive.Delete NuxeoDrive.FileSystemItemExists NuxeoDrive.GenerateConflictedItemName NuxeoDrive.GetRoots NuxeoDrive.GetChangeSummary NuxeoDrive.GetChildren NuxeoDrive.GetClientUpdateInfo NuxeoDrive.GetFileSystemItem NuxeoDrive.GetTopLevelFolder NuxeoDrive.GetTopLevelChildren NuxeoDrive.Move NuxeoDrive.SetSynchronization NuxeoDrive.Rename NuxeoDrive.SetVersioningOptions NuxeoDrive.SetupIntegrationTests NuxeoDrive.TearDownIntegrationTests NuxeoDrive.UpdateFile NuxeoDrive.WaitForElasticsearchCompletion NuxeoDrive.WaitForAsyncCompletion Repository.PageProvider Context.PopDocument Context.PopDocumentList Context.PopBlob Context.PopBlobList Document.PublishToSection Context.PullDocument Context.PullDocumentList Context.PullBlob Context.PullBlobList Context.PushDocument Context.PushDocumentList Context.PushBlob Context.PushBlobList WebUI.AddToClipboard WebUI.PushDocumentToSeamContext WebUI.AddToWorklist LocalConfiguration.PutSimpleConfigurationParameters LocalConfiguration.PutSimpleConfigurationParameter Repository.Query Audit.Query Repository.ResultSetPageProvider WebUI.RaiseSeamEvents Blob.ReadMetadata Context.SetMetadataFromBlob Directory.ReadEntries WebUI.Refresh WebUI.Refresh Document.RemoveACL Services.RemoveDocumentTags Document.RemoveEntryOfMultivaluedProperty Blob.RemoveFromDocument Document.RemovePermission Document.RemoveProperty Collection.RemoveFromCollection Render.Document Render.DocumentFeed TemplateProcessor.Render Document.ReplacePermission Document.Reload Picture.Resize Context.RestoreDocumentInput Context.RestoreDocumentsInput Context.RestoreBlobInput Context.RestoreBlobsInput Document.RestoreVersion Context.RestoreBlobInputFromScript Context.RestoreDocumentInputFromScript Context.RestoreDocumentsInputFromScript Repository.ResultSetQuery Document.Routing.Resume.Step Workflow.ResumeNode Counters.GET RunOperation RunDocumentOperation Context.RunDocumentOperationInNewTx RunFileOperation RunOperationOnList RunOperationOnProvider RunOperationOnListInNewTx RunInputScript RunScript WebUI.RunOperationInSeam Document.Save Seam.SaveDocumentInUI Repository.SaveSession SeamActions.GET Document.Mail Event.Fire Document.AddACE Context.SetVar Context.SetInputAsVar LocalConfiguration.SetSimpleConfigurationParameterAsVar Document.Routing.SetRunningStepFromTask Document.SetBlob Document.SetBlobName WebUI.SetJSFOutcome Workflow.SetNodeVariable Document.Routing.Step.Done Document.Routing.BackToReady Document.Routing.EvaluateCondition Context.SetWorkflowVar WebUI.ShowCreateForm Document.CreateVersion Context.StartWorkflow Search.SuggestersLauncher Services.TagDocument Traces.Get Traces.ToggleRecording Document.SetMetadataFromBlob Seam.GetChangeableDocument Seam.FetchFromClipboard Seam.GetCurrentDocument Seam.GetCurrentDocumentFromWorkflow Seam.GetCurrentWorkflowInstance Seam.GetDocumentsFromSelectionList Seam.FetchFromWorklist Document.Unlock Document.UploadFromURL Document.Update Document.SetProperty Document.Routing.UpdateCommentsInfoOnDocument Directory.UpdateEntries Workflow.UserTaskPageProvider VersionAndAttachFile VersionAndAttachFiles Blob.SetMetadataFromDocument Blob.SetMetadataFromContext Blob.CreateZip acceptComment addCurrentDocumentToWorklist blobToPDF cancelWorkflow conditionalTask decideNextStepAndSimpleValidate downloadFilesZip evaluateCondition followLifeCycleTransition followLifeCycleTransitionTask initInitiatorComment logInAudit nextAssignee notifyInitiatorEndOfWorkflow publishDocument publishTask reinitAssigneeComment rejectComment Workflow.RemoveRoutingTask sendTaskCreatedNotificationMail setDone setNextStep setTaskDone simpleChooseNextOption1AndDone simpleChooseNextOption2AndDone simpleRefuse simpleTask simpleUndo simpleValidate terminateWorkflow undoRunningTask updateCommentsOnDoc validateDocument voidChain xmlExportRendition zipTreeExportRendition

Workflow.CreateRoutingTask

Favorite.GetDocuments

Blob.ToPDF

Image.Blob.Resize

Document.AddRelation

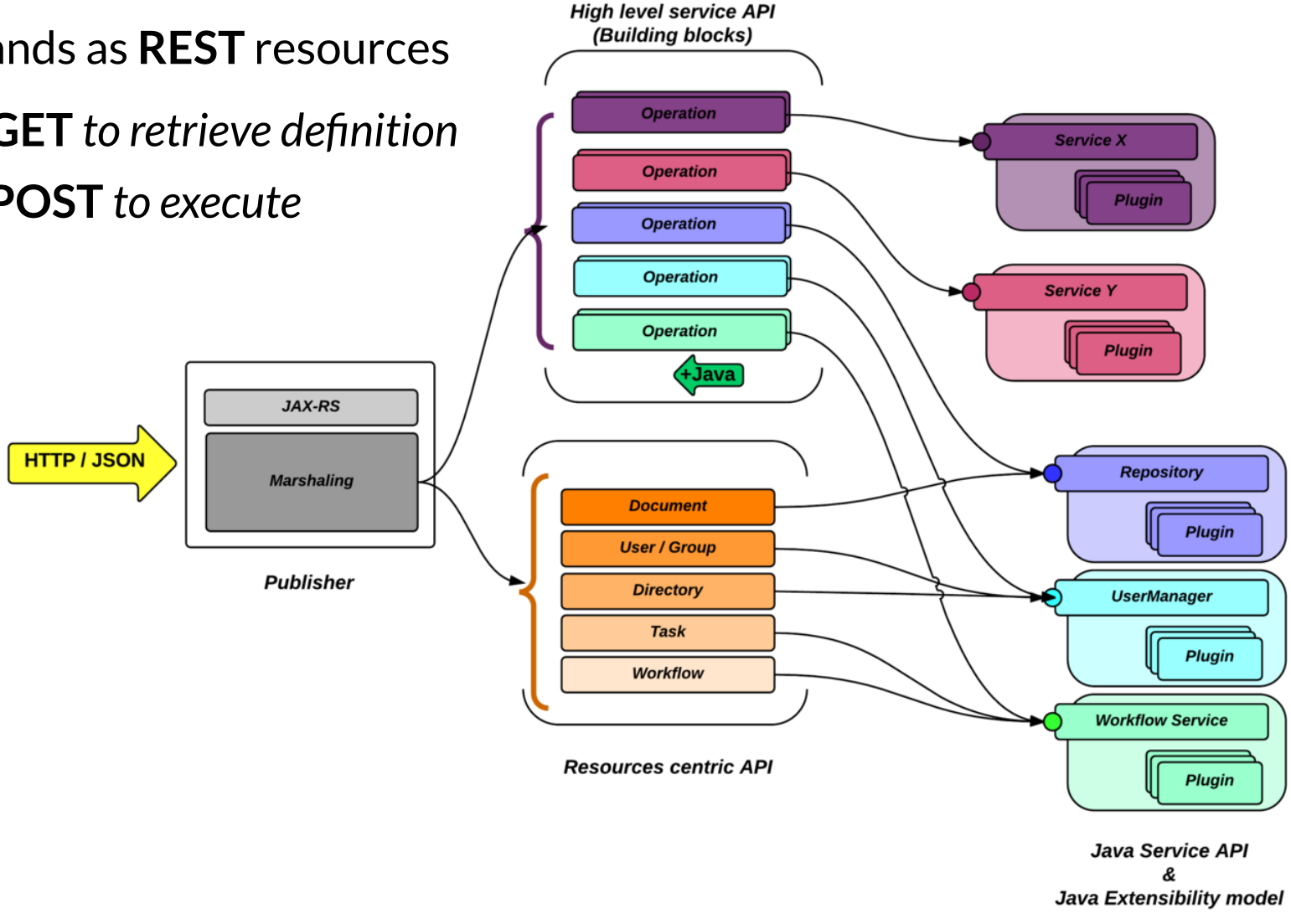
lot of contributed operations

PRINCIPLES

nuxeo

Commands as **REST** resources

- **GET** *to retrieve definition*
- **POST** *to execute*

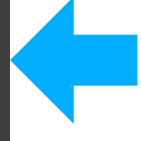


GET AN OPERATION

nuxeo

```
GET /nuxeo/api/v1/automation/Document.PageProvider HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: application/json
{
    "id":"Document.PageProvider",
    "label":"PageProvider",
    "description":"Perform a query ...",
    "signature":[ "void", "documents" ],
    "params":[
      {  "name":"page",
         "type":"integer",
         "required":false
      },{
         "name":"query",
         "type":"string",
         "required":false, },
      ... ]
}
```



GET AN OPERATION

Operation *Repository.PageProvider* (PageProvider)

Description

Perform a query or a named provider query on the repository. Result is paginated. The query result will become the input for the next operation. If no query or provider name is given, a query returning all the documents that the user has access to will be executed.

General information

Operation id:	Repository.PageProvider
Aliases:	[Document.PageProvider]
Category:	Fetch
Label:	PageProvider
Requires:	
Since:	

Parameters

Name	Description	Type	Required	Default value
currentPageIndex		integer	no	
documentLinkBuilder		string	no	
language		string	no	NXQL
maxResults		string	no	
namedParameters	Named parameters to pass to the page provider to fill in query variables.	properties	no	
page		integer	no	
pageSize		integer	no	
providerName		string	no	
query		string	no	
queryParams		stringlist	no	
sortBy	Sort by properties (separated by comma)	string	no	
sortInfo		stringlist	no	
sortOrder	Sort order, ASC or DESC	string	no	ASC, DESC

Signature

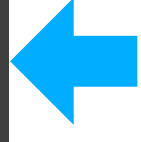
Inputs:	void
Outputs:	documents

RUN AN OPERATION

nuxeo

```
POST /nuxeo/api/v1/automation/Document.PageProvider HTTP/1.1
Content-Type: application/json+nxrequest
{ "params" :
    { "query" : "select * from Note",
      "page" : 0
    }
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "entity-type": "documents",
  "pageIndex": 0,
  "pageSize": 2,
  "pageCount": 2,
  "entries": [
    {
      "entity-type": "document",
      "repository": "default",
      "uid": "3f76a415-ad73-4522-9450-d12af25b7fb4",
      ...
    }, { ...}, ...
 ]
}
```



RESOURCES & AUTOMATION

nuxeo

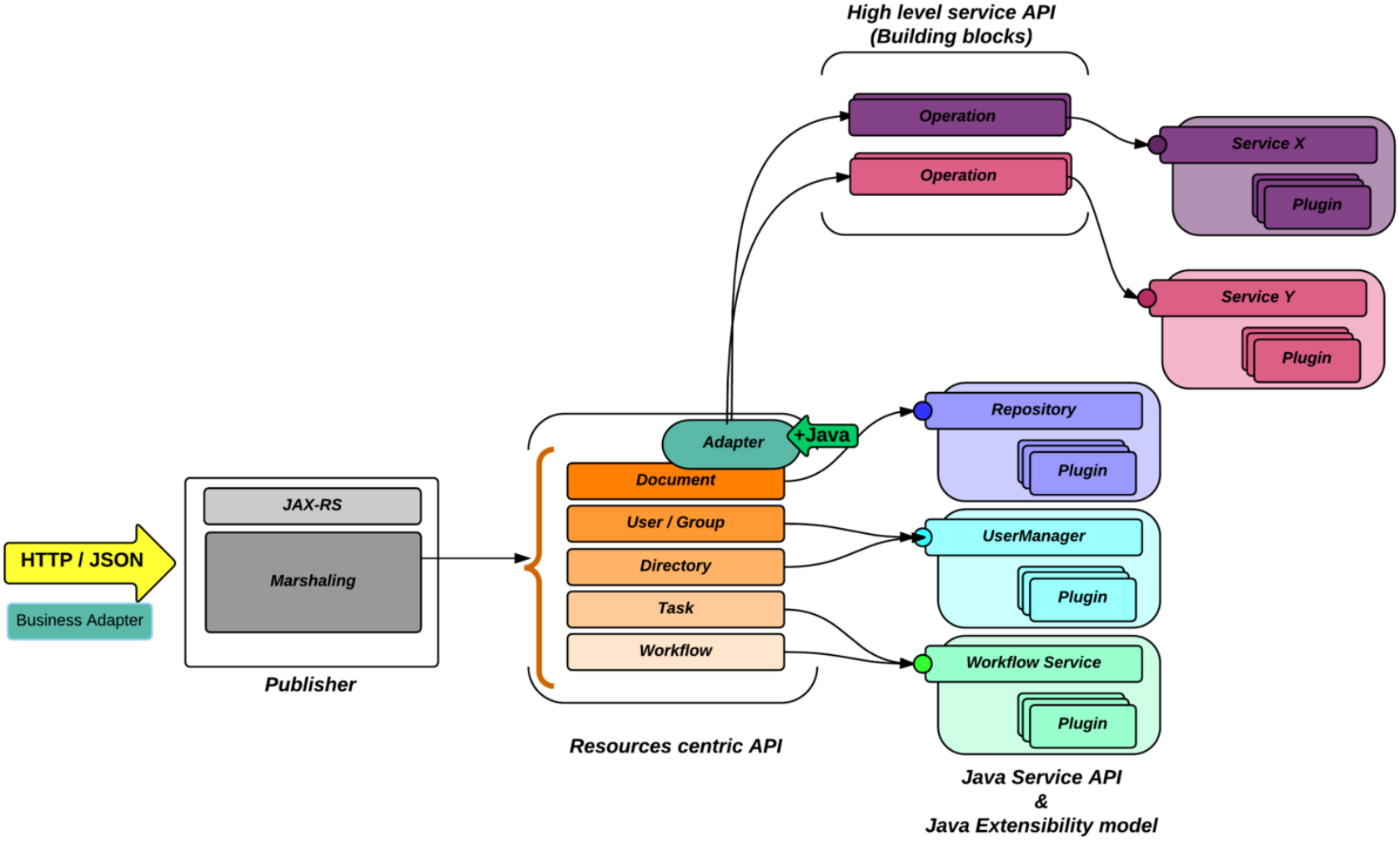
- Share the **marshaling layer** and **extension**
  - *Enrichers, Resolvers are available too*

- **Compose** Resources and Automation API
  - *Pipe Resources as input for Automation Operation*

```
> cat /doc/path/somedoc | command(p3,p4)
```

RESOURCES & AUTOMATION

nuxeo

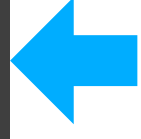


RESOURCES > AUTOMATION

nuxeo

```
POST /nuxeo/api/v1/path/somePath/@op/Blob.ToPDF HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: application/pdf
...
```



ARE WE HAPPY WITH THAT ?

nuxeo

- **Efficiency**

- **Flexibility**

- **Coverage**
  - *all services and plugins can contribute*

- **Extensibility**
  - *good, but limited to Java Developers*

- **Consistency**
  - *still no way to align Application Transactions*

nuxeo

MORE COMPOSITION

assemble API blocks without having to code

build business API

nuxeo

COMPOSABLE API

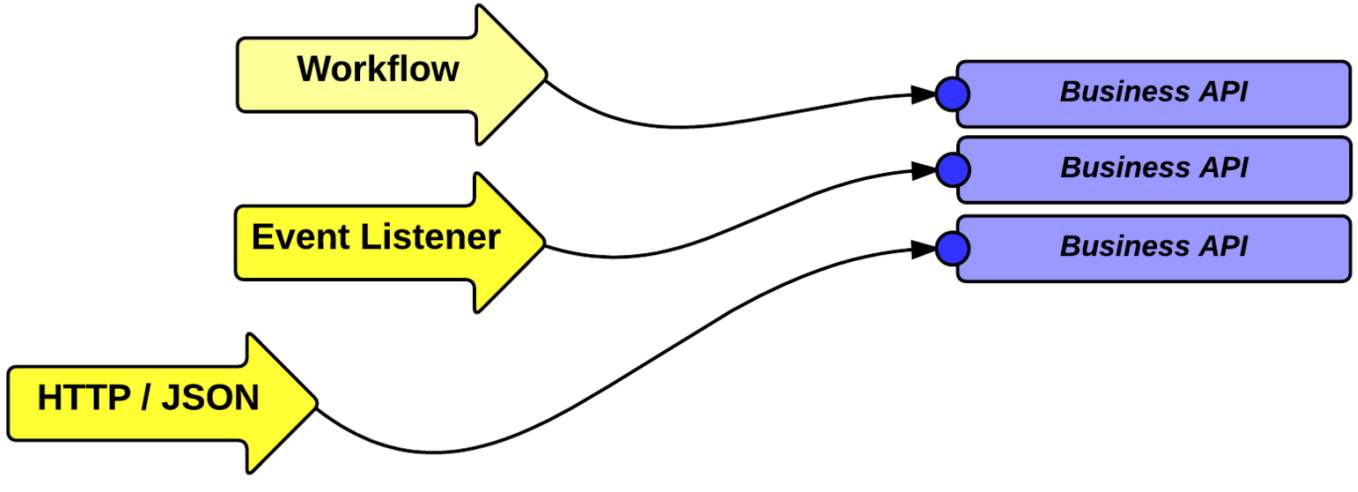
Expose the API that matches client needs

GOALS

nuxeo

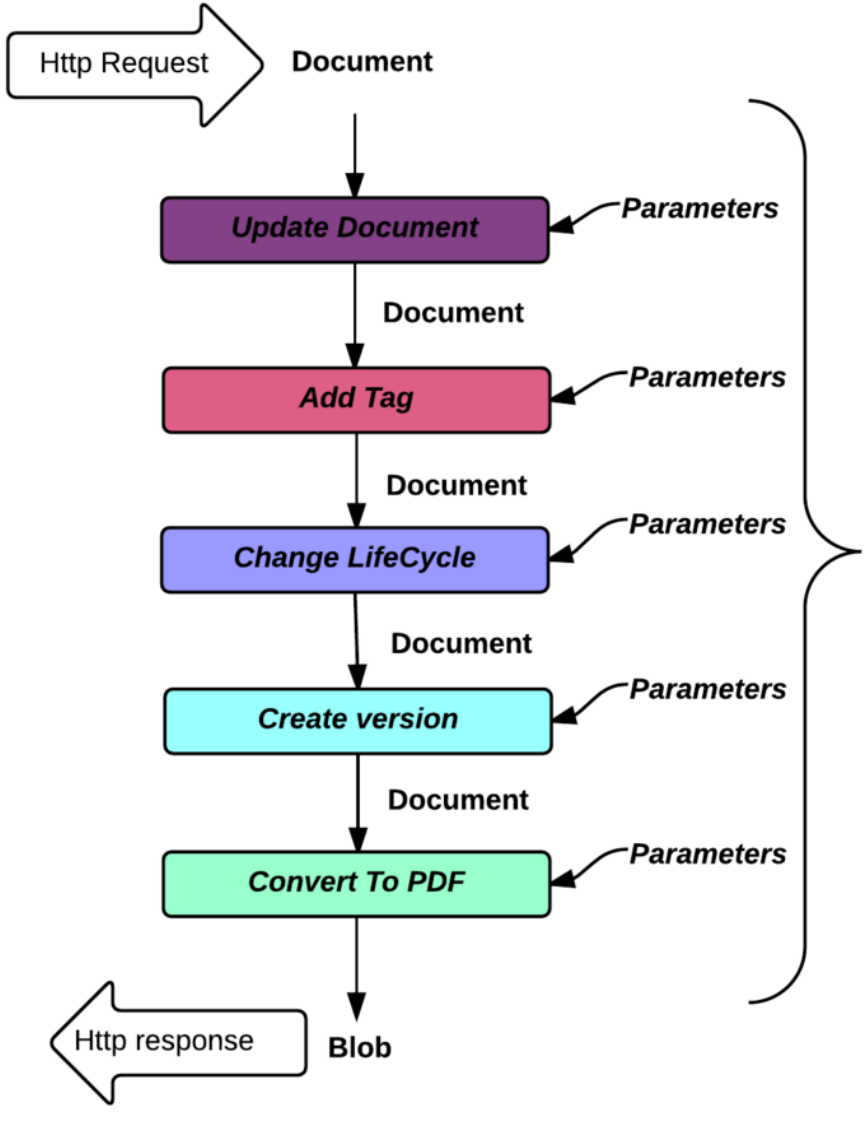
- **Tailor the API** to match application requirements
  - *one API behind every action / button*

- Allow **business analysts** or **UI developers** to tailor the API
  - define what API is exposed
    - UI & Workflow needs



AUTOMATION CHAIN

nuxeo



- **Assemble** operations in a **chain**
- **Pipe** Output / Input
- Give it a **name**
- Call and execute within a **single transaction**

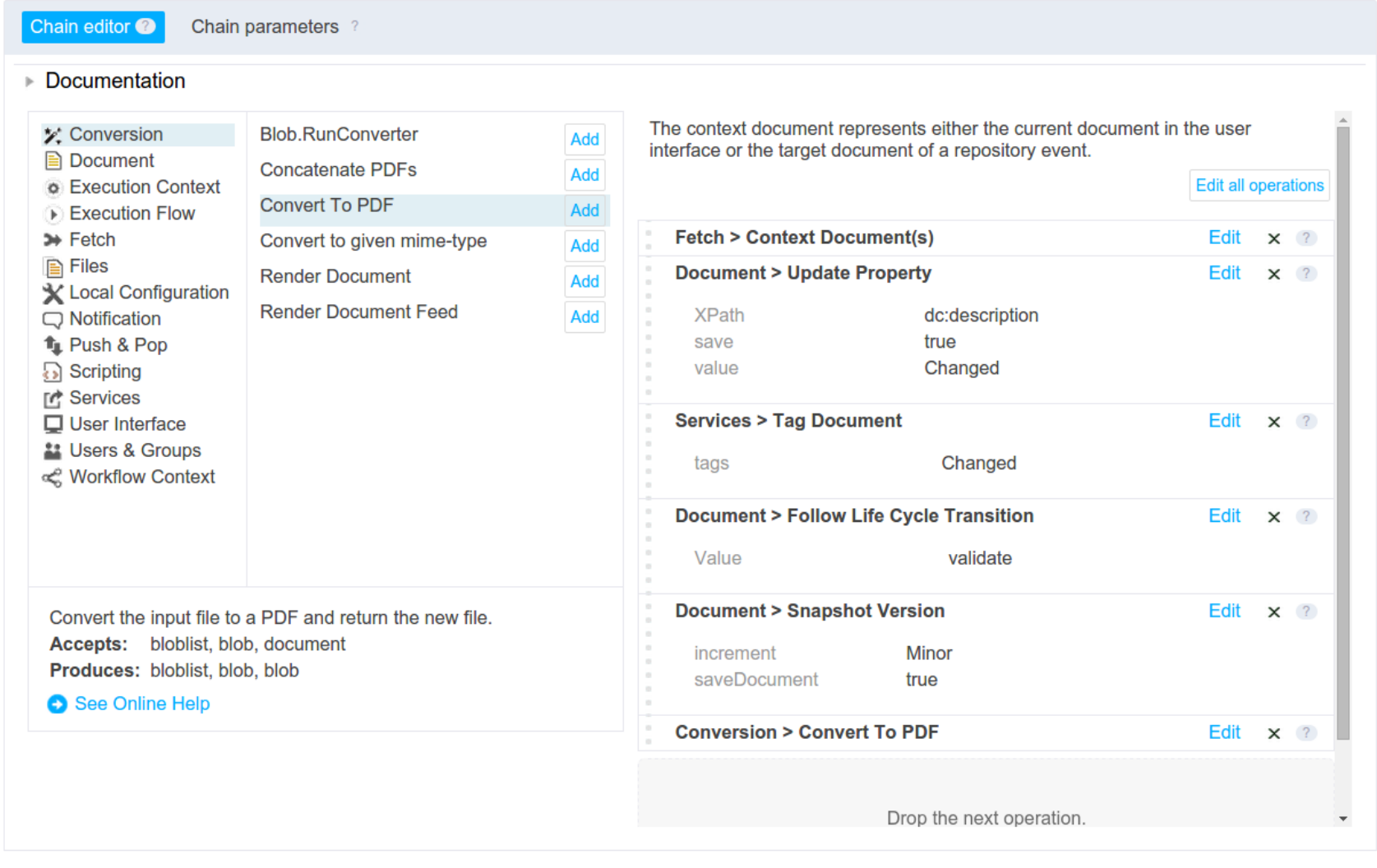
One Transaction

One Context

Server side assembly

ASSEMBLING CHAINS

nuxeo

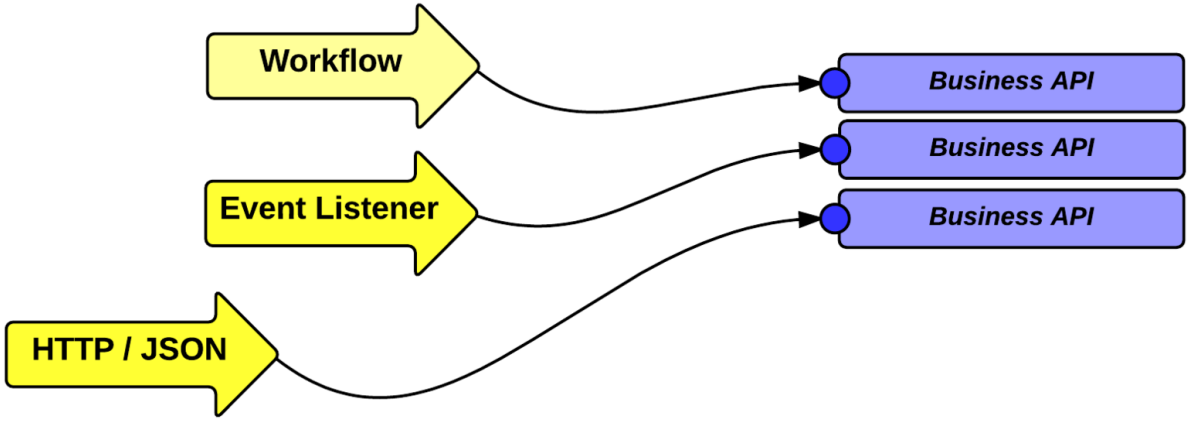


IT DOES WORK !

nuxeo

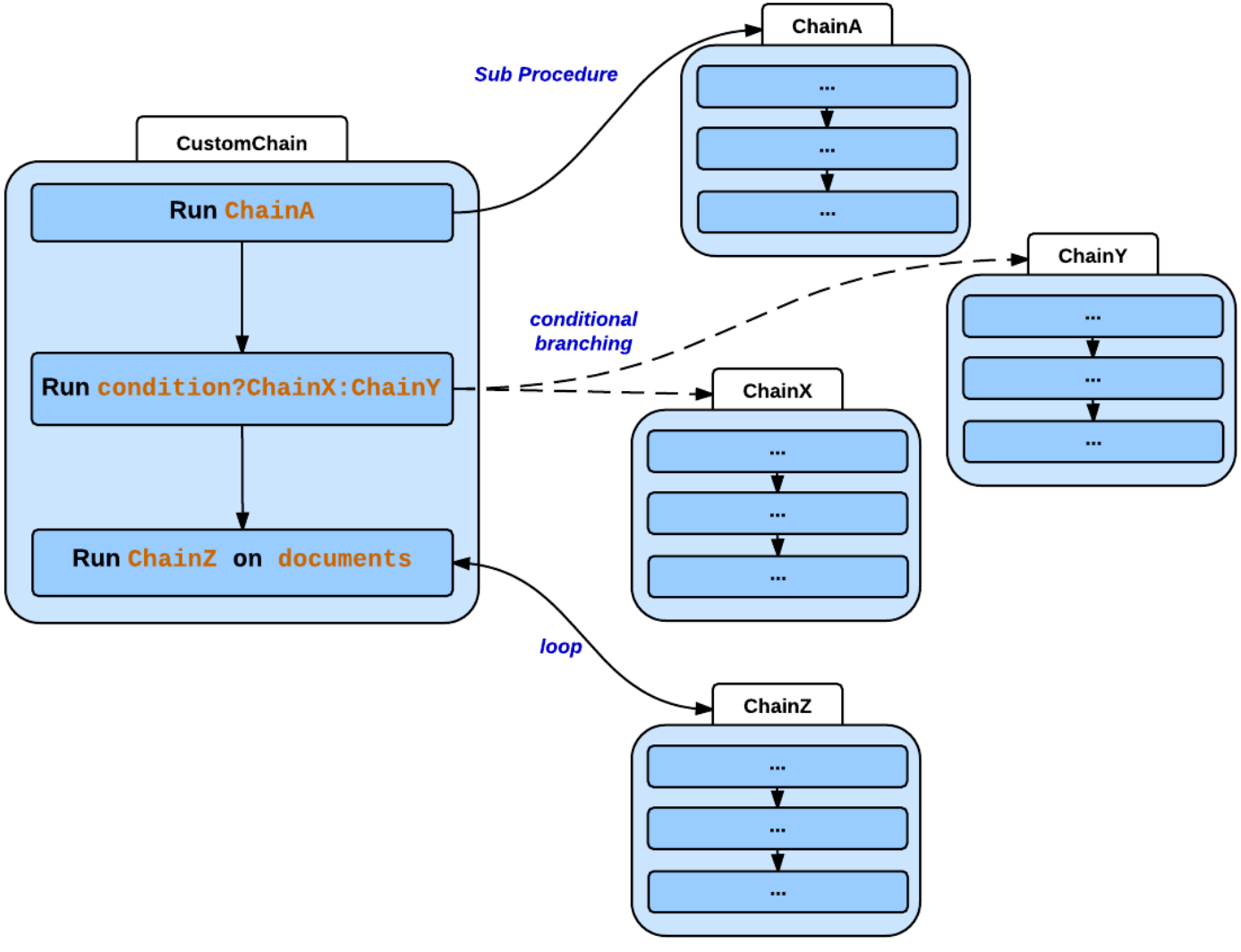
- **Business users & Front end developers leverage this**
  - *to expose custom API for their UI*
  - *to build custom logic inside their Workflows*
  - *to add automatic processing (listeners)*

    - Actually it **works** almost **too well**
      - *users do awfully complicated things*
      - *chains calling chains calling chains ...*



HAVE WE CREATED A MONSTER ?

nuxeo



nuxeo

GO FURTHER THAN THE CHAIN MODEL

USE NASHORN TO ASSEMBLE OPERATIONS

- *Operations remain the building blocks*
- *JavaScript is the glue code to assemble them*

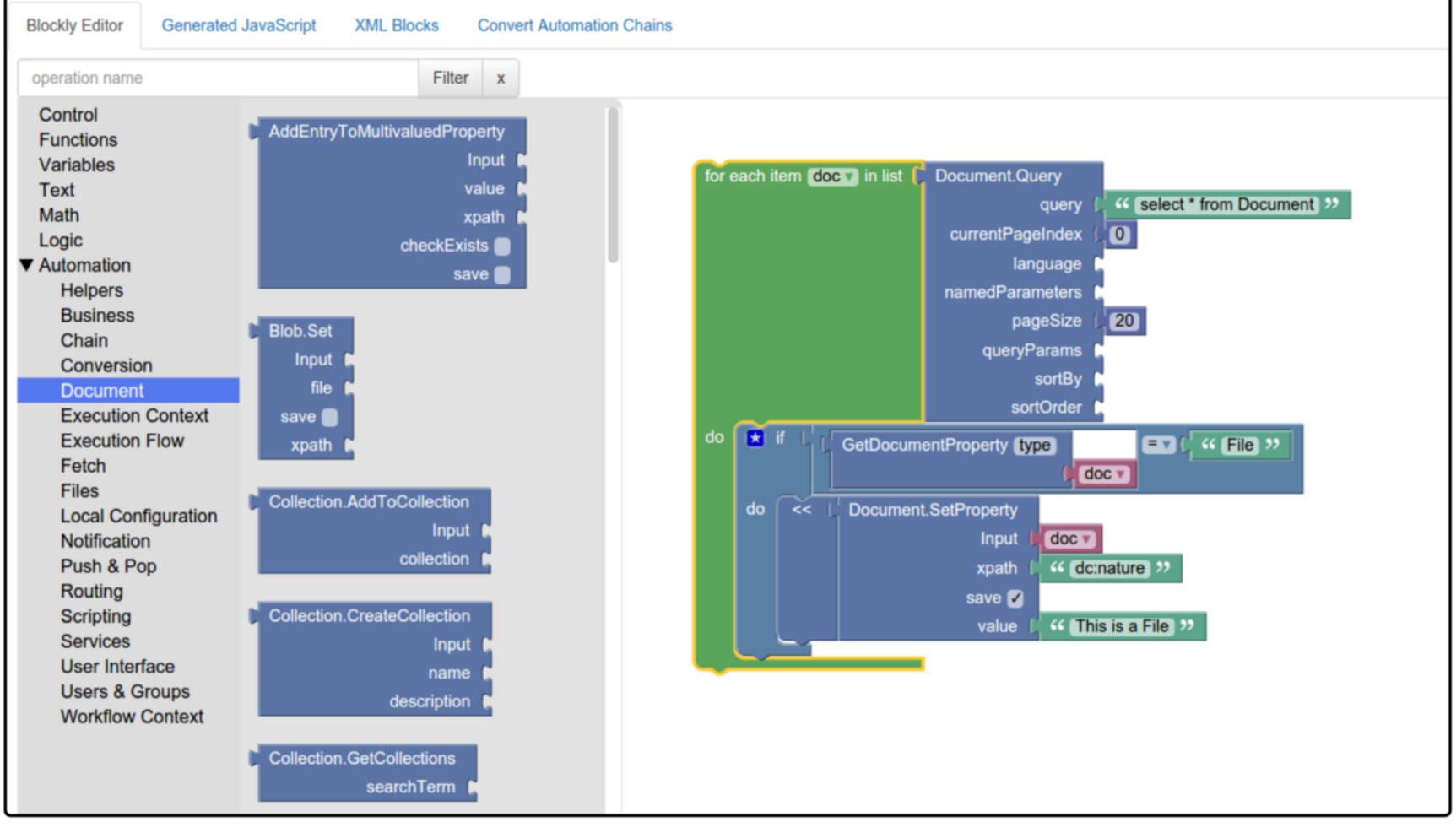
better control of the flow

```
function run(input, params) {

  var docs = Seam.GetSelectedDocuments(input, {
});
/* Description: Fetch the documents selected in the current folder listing */

  if(docs.length>3){
    var index;
    for(index=0;index<docs.length;++index){
      Document.SetProperty(input, {
        /*required:true - type: string*/
        'xpath': "dc:title",
        /*required:false - type: boolean*/
        'save': true,
        /*required:false - type: serializable*/
        'value': "test"
      });
      WebUI.Refresh(input, {});
    }
  }else{
      WebUI.AddMessage(input, {
        /*required:true - type: string*/
        'message': "DISPLAY IT",
        /*required:true - type: string*/
        'severity': "WARN"
      });
  }
}
```

Document.

```
Document.AddACL
Document.AddToCollection
Document.CheckIn
Document.CheckOut
Document.Copy
Document.Create
Document.CreateLiveProxy
Document.AddRelation
Document.Delete
Document.DeleteRelation
Document.Export
Document.FetchByProperty
Document.Filter
Document.FollowLifecycleTransi
Document.GetBlobs
Document.GetChild
Document.GetChildren
```

Nuxeo Automation Scripting : Blockly Editor



ARE WE HAPPY WITH THAT ?

nuxeo

- Efficiency
- Flexibility
- Coverage
- Extensibility
- Consistency

nuxeo

DYNAMIC AND COMPOSABLE API

What about the trade-off ?

DYNAMIC API COMES WITH A PRICE

nuxeo

- **Documentation** is a challenge
  - *maintain up to date and exhaustive*
- **Clients** needs to deal with the dynamic aspect
  - *data mapping*
  - *discover APIs*
- **Debugging** API calls can be challenging

Introspection is a requirement

INTROSPECTION

nuxeo

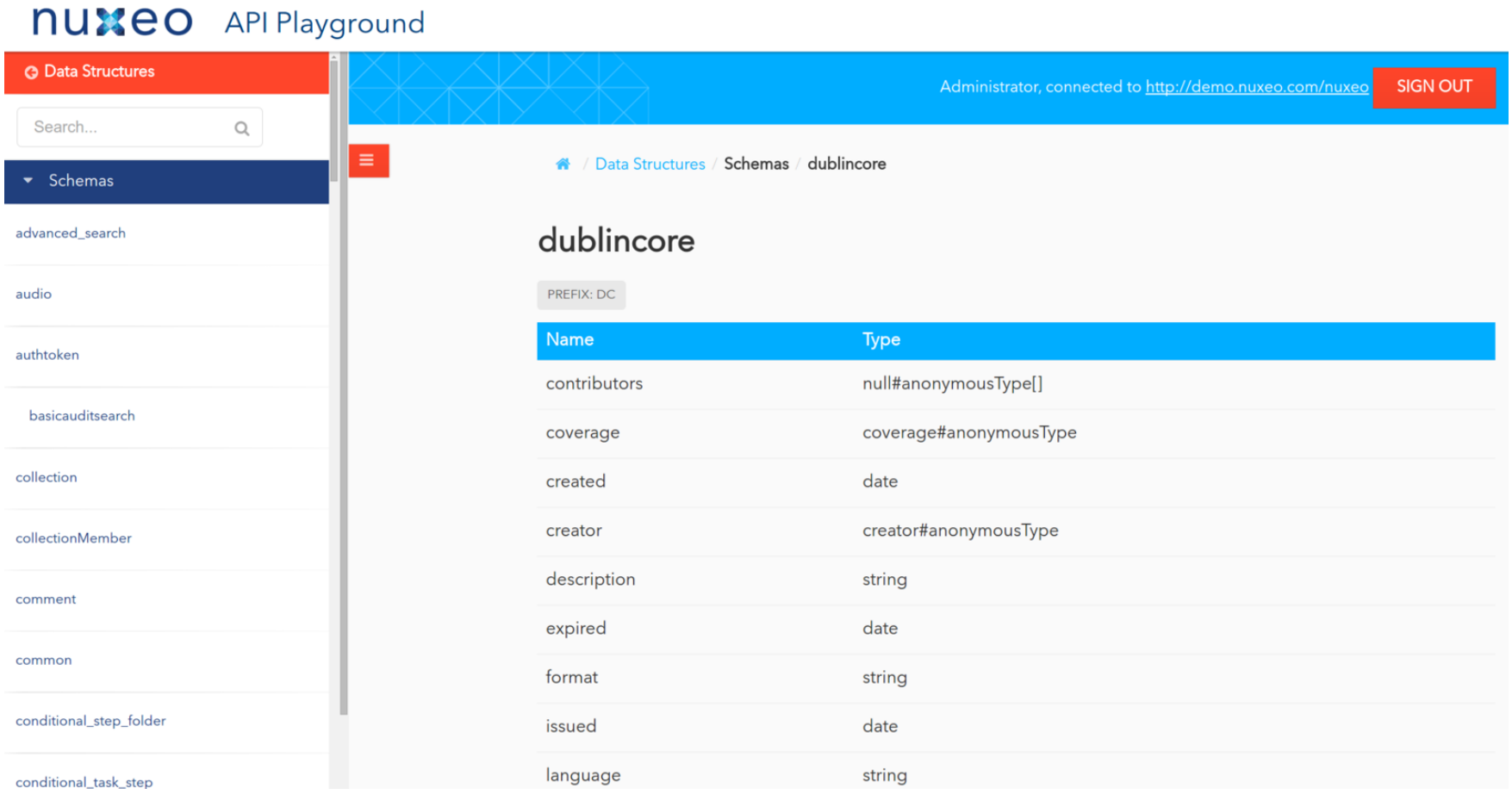
- **Trace** feature
  - *Allow client to retrieve traces of nested executions* *(dev mode)*
- **Introspect** the Command API
  - GET *retrieve definition + doc*
- Add REST API to **introspect configuration resources**
  - *Document types, Schemas*
  - *(Widgets & Forms)*

➡ Provide a **Playground** to test

- *connect to a remote server*
- *introspect server API and structures*

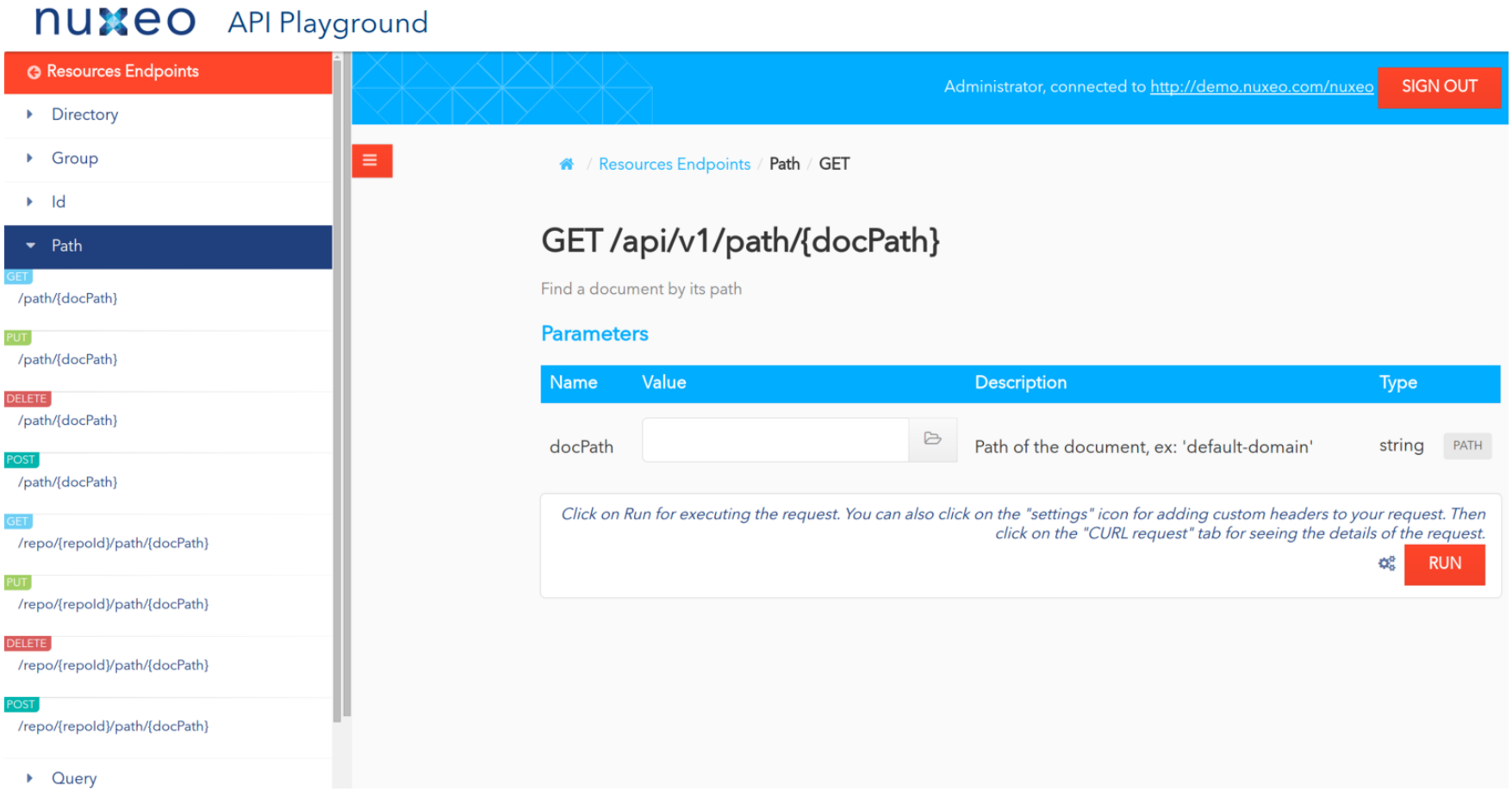
Introspect Data Structures

http://nuxeo.github.io/api-playground/



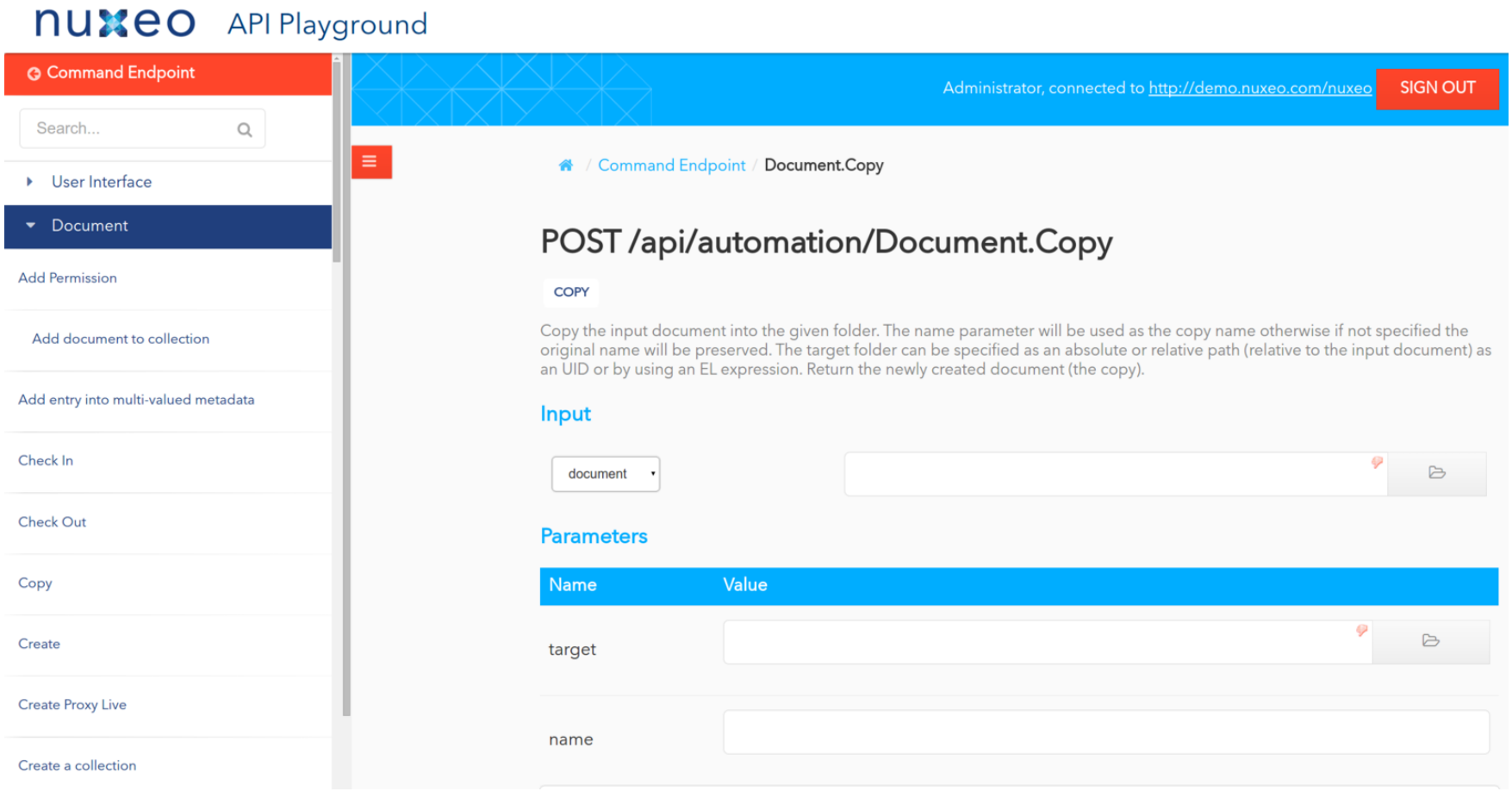
Introspect Resources

http://nuxeo.github.io/api-playground/



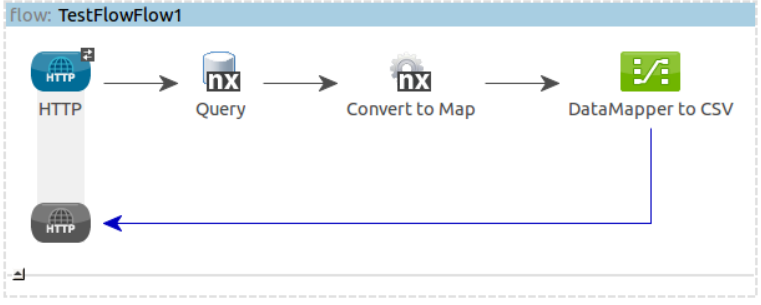
Introspect Command API

http://nuxeo.github.io/api-playground/

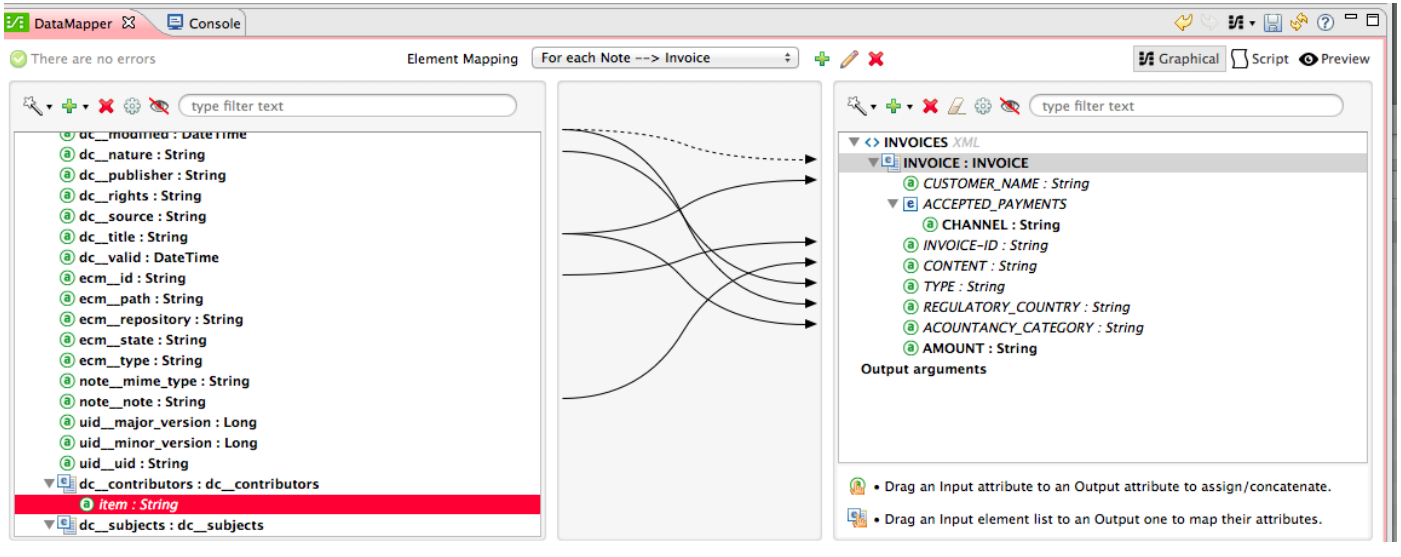


INTROSPECTION IN ACTION

nuxeo



muleESB



Using introspection from Mule ESB DataSense

CLIENTS LIBS

nuxeo

Dynamic API ➜ more complex to use !?

client libraries

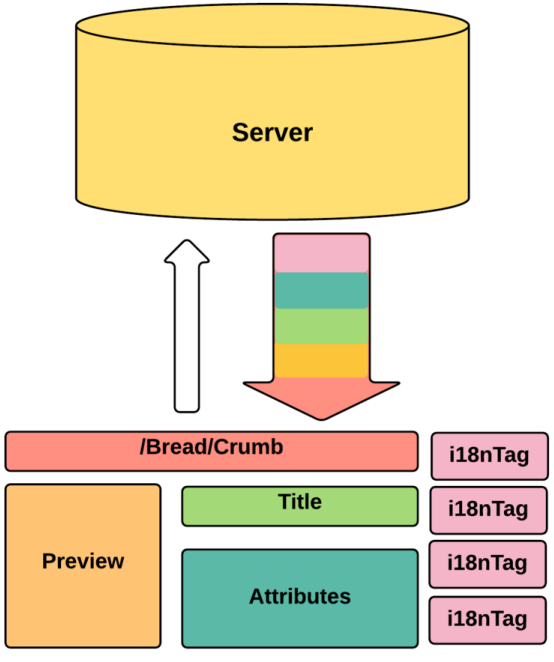
nuxeo

NEXT CHALLENGES

UI INTROSPECTION

- **Introspect UI** components tree
  - *gather needed data*

- **Auto-configure** Nuxeo Calls
  - *schemas, enrichers, resolvers ...*
- **Post back** aggregated data
  - *reverse enrichers and resolvers ?*



SOME LINKS

nuxeo

http://nuxeo.github.io/api-playground/

https://university.nuxeo.io/

https://doc.nuxeo.com/display/NXDOC/REST+API

https://github.com/nuxeo

nuxeo-workshop-restapi

nuxeo

ANY QUESTIONS ?

Thank You !

http://www.nuxeo.com/careers/