

# REST

13th November 2013



For customers or partners with NDA



- Definitions of Wikipedia
  - REST
    - REST (Representational state transfer) is a style of software architecture for distributed hypermedia systems such as the World Wide Web
    - REST-style architectures consist of clients and servers. Clients initiate requests to servers; servers process requests and return appropriate responses. Requests and responses are built around the transfer of representations of resources. A resource can be essentially any coherent and meaningful concept that may be addressed. A representation of a resource is typically a document that captures the current or intended state of a resource.

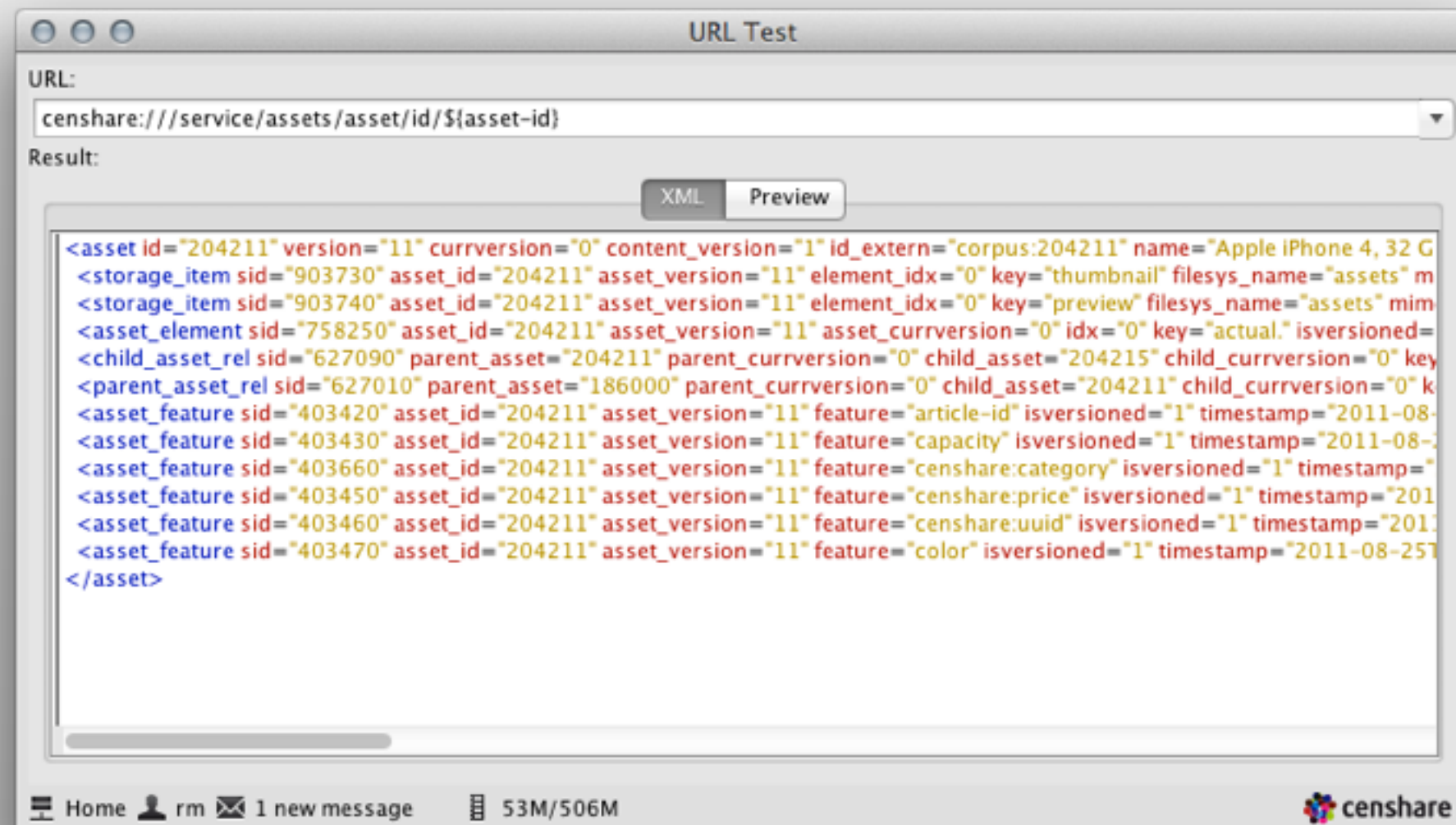
# Overview



- censhare
  - censhare has implemented a REST interface to have access over HTTP requests to
    - Assets
      - Search
      - XML
      - Icon
      - Files
      - Result of transformations
    - Resources
      - Icons
    - Temporary file systems
      - Files
  - This presentation describes the functionality of censhare versions 4.5.14, 4.6.9 and 4.7.2 or higher

# Overview

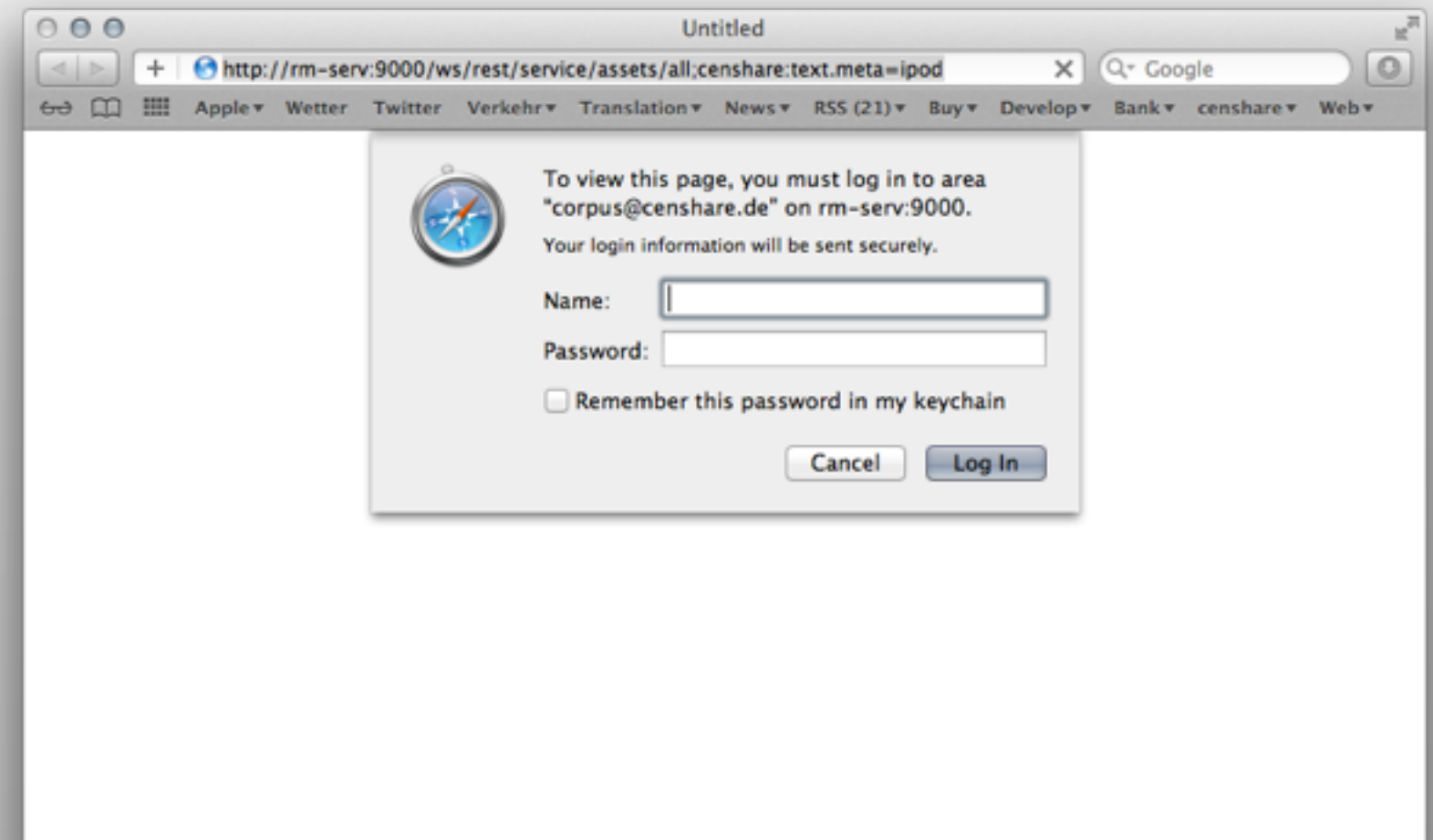
- censhare
  - Provides a test window in all censhare Clients in admin mode with menu item „URL Test“
  - ID of current asset selection can be used in URL with placeholder „\${asset-id}“, if one asset is selected



- Interfaces
  - Internal (censhare Server and censhare Clients)
    - Is always enabled and has not to be configured
    - URL of censhare Server and censhare Client starts with „censhare:///service/“
    - URL of censhare Web-Client starts with relative path „webclient/url/“
    - Use „system-property('censhare:url-prefix')“ to get correct value for Client or Web-Client
  - External (Access over HTTP)
    - REST service is provided by the „HTTP Server“ service
      - Embedded web server based on Jetty on each censhare application server
    - Is disabled per default and has to be enabled
    - Port is „9000“ as default and can be modified at file „jetty.xml“ in folder „httpserver“ of folder „services“
    - Access over URL starting with „http://[host]:[port]/ws/rest/service/“

# Permissions

- Interfaces
  - Internal (censhare Clients)
    - Requests are execute from the logged-in user
    - Only the information is returned to the logged-in user has access
  - External (Access over HTTP)
    - User has to login with censhare user and password information
    - Only the information is returned to the logged-in user has access
    - You may need to use „https“ protocol for more security



# Communication



- External (Access over HTTP)
  - Implemented requests methods in censhare 4.3
    - GET
      - All parameters are transferred in the URL
    - POST
      - Submits data to be processed

- Part „search“
  - Syntax
    - Search items are separated with „;“
    - Search items are combined as „and“ search
    - Search items consists of index name of censhare database + „=“ + search value
    - If search items contains one or more „\*“ then a like search is executed
  - Examples
    - „censhare:asset.name=test.txt“ search for assets whose name is „test.txt“
    - „censhare:resource-key=transform:product“ search for assets whose resource key is „transform:product“
    - „censhare:asset.name=test.txt;censhare:asset.type=text.“ search for assets whose name is „test.txt“ and type is text



# Parameter



- Part „asset“
  - „/asset/id/“ + asset ID
    - and „/version/“ + asset version
    - or „/currversion/“ + asset current version
  - „/asset/uuid/“ + asset UUID
    - and „/version/“ + asset version
    - or „/currversion/“ + asset current version
  - „asset;“ + [search parameter part]

# Parameter



- Part „element“
  - „/element/“ + [target] or [actual]
    - and „/“ + [element index]
    - or „/page/“ + [page number starting with „0“ for first page]

# Parameter



- Part „storage“
  - „/storage/“ + storage key
    - Storage key is e.g. „master“, „preview“, „thumbnail“, ...

# Parameter

- Part „icon“
  - „/iconset/“ + name of iconset
    - Optional, default is „default“
  - and „/background/“ + name of the background
    - Optional, „dark“ or „bright“ („dark“ is default)
  - and „/minsize/“ + size in pixel
    - Optional, default is „16“

- Get asset XML
  - Returns XML of asset of given parameters
  - Syntax
    - „assets/“ + [asset parameter part]
  - Examples
    - „assets/asset/id/123“ returns XML of asset ID „123“
    - „assets/asset/id/123/version/1“ returns XML of version „1“ of asset ID „123“
    - „assets/asset/id/123/currversion/-2“ returns XML of checked out version of asset ID „123“
    - „assets/asset;censhare:asset.id=123;censhare:asset.currversion=-2“ returns XML of checked out version of asset ID „123“
    - „assets/asset;censhare:resource-key=transform:product“ returns XML of current version of first asset with resource key „transform:product“

- Get files

- Returns file with given parameters of asset with given parameters

- Syntax

- „assets/“ + [asset parameter part] + „/“ + [optional element part] + „/“ + [storage parameter part] + „/file“

- Examples

- „assets/asset/id/123/storage/master/file“ returns master file of asset ID „123“
- „assets/asset/id/123/storage/preview/file“ returns preview file of asset ID „123“
- „assets/asset/id/123/element/actual/page/1/storage/preview/file“ returns preview of first page of asset ID „123“
- „assets/asset/id/123/element/actual/0/storage/preview/file“ returns preview of element index 0 of asset ID „123“
- „assets/asset;censhare:resource-key=transform:product/storage/master/file“ returns master file of first asset with resource key „transform:product“

# URLs



- Get files
  - Examples
    - „assets/asset/id/123/storage/master/file;container=zip/docProps/thumbnail.jpeg“ returns file „thumbnail.jpeg“ of folder „docProps“ of master ZIP file of asset ID „123“

- Get asset icon description
  - Returns description of asset icon with given parameters
  - Syntax:
    - „assets/“ + [asset parameter part] + „/icon/“ + [icon parameter part]
  - Examples
    - „assets/asset/id/123/icon“ returns asset icon of asset ID „123“
    - „assets/asset/id/123/version/1/icon“ returns asset icon of version 1 of asset ID „123“
    - „assets/asset/id/123/currversion/-2/icon“ returns asset icon of checked-out version of asset ID „123“
    - „assets/asset/id/123/icon/iconset/default“ returns asset icon of default iconset of asset ID „123“
    - „assets/asset/id/123/icon/iconset/default/background/dark“ returns asset icon of default iconset for dark background look&feel of asset ID „123“



- Get asset icon file
  - Returns asset icon with given parameters
  - Syntax:
    - „assets/“ + [asset parameter part] + „/icon/“ + [icon parameter part] + „/file“
  - Examples
    - „assets/asset/id/123/icon/file“ returns asset icon of asset ID „123“
    - „assets/asset/id/123/version/1/icon/file“ returns asset icon of version 1 of asset ID „123“
    - „assets/asset/id/123/currversion/-2/icon/file“ returns asset icon of checked-out version of asset ID „123“
    - „assets/asset/id/123/icon/iconset/default/file“ returns asset icon of default iconset of asset ID „123“
    - „assets/asset/id/123/icon/iconset/default/background/dark/file“ returns asset icon of default iconset for dark background look&feel of asset ID „123“

- Get result of asset transformations
  - Returns result of transformation of asset with given parameters
  - Parameters
    - Standard parameters
      - „asset-id“: Asset ID of the transformation
      - „key“: Transformation key
    - Optional parameters can be added
    - Parameters are available with the given names in XSLT
      - e.g. „<param name="asset-id"/>“
  - Syntax
    - „assets/“ + [asset parameter part] + „/transform;key=“ + [transformation key] + [„;“ + transformation parameter]

- Get result of asset transformations
  - Examples
    - „assets/asset/id/123/transform;key=test;format=xml“ returns result of transformation with key „test“ and additional parameter „format“ with value „xml“
    - „assets/asset/id/123/transform;key=test;format=xml;param1=a;param2=b“ returns result of transformation with key „test“ and additional parameter. Following parameters are available in XSLT
      - „<param name="asset-id"/>“ with value „123“
      - „<param name="key"/>“ with value „test“
      - „<param name="format"/>“ with value „xml“
      - „<param name="param1"/>“ with value „a“
      - „<param name="param2"/>“ with value „b“

- Get result of asset transformations
  - Localization with transformations
    - The transformation key „localize“ is used to localize the XML
    - Placeholder, e.g. „`{asset}`“ is converted to „Asset“ with global properties
    - Example
      - `censhare:///service/assets/asset/id/205371/transform;key=localize`

- Get directory of ZIP files
  - Returns a list of all files of ZIP files as XML
  - Can be added to all REST URLs based on file results
  - Syntax
    - [REST URL to ZIP file] + „;container=zip“
  - Example
    - „assets/asset/id/123/storage/master/file;container=zip“
    - „filesystem/assets/13/89/138931.zip;container=zip“
- Example result

```
<entries>
  <entry name="test/" />
  <entry name="test/test.pdf" />
  <entry name="test1.jpg" />
  <entry name="test2.jpg" />
</entries>
```

- Get files of ZIP files
  - Returns defined files of ZIP files
  - Can be added to all REST URLs based on file results
  - Relative path is separated with „/“
  - Syntax
    - [REST URL to ZIP file] + „;container=zip/“ + [file path]
  - Example
    - „assets/asset/id/123/storage/master/file;container=zip/docProps/thumbnail.jpeg“
    - „filesystem/assets/13/89/138931.zip;container=zip/designmap.xml“

- Get icon
  - Returns asset icon with given parameters
  - Syntax
    - „resources/icon/“ + [icon key] + „/“ + [icon parameter part] + „/file“
  - Examples
    - „resources/icon/add/file“ returns icon with key „add“
    - „resources/icon/add/iconset/default/file“ returns icon of default iconset with key „add“
    - „resources/icon/add/iconset/default/background/dark/file“ returns icon of default iconset for dark look&feel with key „add“

- Search for asset XML (GET request)
  - Returns XML of any assets that matches with given search criterias
  - Syntax
    - „assets/all;“ + [search parameter part]
  - Examples
    - „assets/all;censhare:asset.type=picture.“ returns every asset XML whose type is „picture“
    - „assets/all;censhare:asset.type=picture.;censhare:text.meta=ipod“ returns every asset XML whose type is „picture“ and meta data contains „ipod“
    - „assets/all;censhare:asset.type=picture.;censhare:asset.name=\*ipod\*/transform;key=test“ returns a transformation with key „test“ of every asset XML whose type is „picture“ and name contains „ipod“



- Search for asset XML (POST request)
  - Returns XML of any assets that matches with given search criterias
  - Syntax
    - URL „assets/query“
    - Client uploads query XML file as POST parameter
      - Query XML file has same syntax as standard asset query requests from client to server
  - Example with „curl“ (Client for URLs)
    - „curl -X POST -T "/Users/rm/censhare/Präsentationen/censhare REST Training/asset-query.xml" -u rm:  
<http://rm-serv:9000/ws/rest/service/assets/query>“
    - „curl“ parameters
      - -X -X/--request <command>
      - -T -T/--upload-file <file>
      - -u -u/--user <user:password>

- Search for asset XML (POST request)
  - Example query XML file
    - Query for assets with „my tasks“ and (asset type is picture or asset type is video)
      - ```
<query count-rows="false" expand-limit="1000" full="200" limit="200" rel-limit="2147483647" start-full="0" start-row="0" type="asset" xmlns:corpus="http://www.censhare.com/xml/3.0.0/corpus">  
  <condition name="censhare:function.my-tasks" op="=" value="1"/>  
  <or>  
    <condition expanded-nodes="picture." name="censhare:asset.type" op="=" partial-load-tree="true" value="picture.*"/>  
    <condition expanded-nodes="video." name="censhare:asset.type" op="=" partial-load-tree="true" value="video.*"/>  
  </or>  
</query>
```

- Get file of file system
  - Returns file of given file system and URL
  - External access is limited to the temporary file system
  - Syntax
    - „filesystem“ + [name of file system] + „/“ + [URL to file]
  - Examples
    - „filesystem/temp/1234456/10000.pdf“ returns file „10000.pdf“ in folder „1234456“ of temporary file system „temp“
    - „filesystem/assets/13/89/138931.jpg“ returns file „138931.jpg“ of the „assets“ file system

# Usage in censhare



- Adobe InDesign
  - URLs for placed files without mounted volumes are REST-URLs (censhare 4.4 and above)



# Usage in censhare



- Adobe InDesign
  - URLs for placed transformation assets are REST-URLs

```
<child_asset_element_rel child_asset="186000" child_currversion="0" child_idx="0"
child_storage_mimetype="application/vnd.adobe.incopy-icml" child_version="11" corpus:dto_flags="pt"
deletion="0" has_placed_attributes="1" id_extern="b591" is_manual_placement="0" is_cancellation="0"
is_locked="0" is_versioned="1" key="actual." overset_charcount="0" overset_linecount="0"
overset_wordcount="0" parent_asset="191000" parent_currversion="0" parent_idx="5" parent_version="25"
rel_idx="0" rotation="0.0" rowid="AAASQFAAGAAABXIAAT" scale_x="1.0" scale_y="1.0" sid="350140"
url="censhare:///service/assets/asset/id/186000/transform;key=transform
%3Aproduct;format=icml;mimetype=application%2Fvnd.adobe.incopy-icml;target-asset-id=191000"
version="0"/>
```

# Usage in censhare



- Content-Editor
  - URLs for placed assets or content-in-content structures are REST-URLs

```
<?xml version="1.0" encoding="UTF-8"?>
<content>
  <title>Biograd na moru</title>
  <text>
    <paragraph>Text</paragraph>
    <image-box align="left" zoom="true">
      <image alt-text="Tests" aspect-ratio="16-9" title="Test"
        xlink:href="censhare:///service/assets/asset/id/21778"
        xmlns:xlink="http://www.w3.org/1999/xlink"/>
      <caption style="style-1">
        <paragraph>Image caption</paragraph>
      </caption>
    </image-box>
  </text>
</content>
```

# Usage in censhare



- Image widget
  - Attribute „url“ of widget „xe:image“ supports REST-URLs (censhare 4.4 and above)

```
<xe:image url="=:concat('censhare:///service/assets/asset/id/', asset/@id, '/storage/thumbnail/file')"/>
```

# Usage in censhare



- HTML widget
  - Attribute „url“ of widget „xe:html“ supports REST-URLs (censhare 4.4 and above)

```
<xe:html url="=:concat('censhare:///service/assets/asset/id/', asset/@id, '/transform;key=transform:icon-name)"/>
```



# Usage in censhare



- Transformations
  - Images in HTML results of transformations are REST-URLs

```
<html>
  <table>
    <tr>
      <td>
        
      </td>
      <td>Apple iPhone 4 (186000)</td>
    </tr>
  </table>
</html>
```

# More infos

- Links
  - Wikipedia – REST
    - <http://en.wikipedia.org/wiki/REST>
  - Principled Design of the Modern Web Architecture
    - <http://www.ics.uci.edu/~taylor/documents/2002-REST-TOIT.pdf>

Thank you

