Purpose:

This API documentation is intended to assist the user with understanding how the CHORUS Funder Dashboard Service delivers aggregated data and paths to actual articles, please see [http://www.chorusaccess.org/services/dashboard-service/](http://www.chorusaccess.org/services/dashboard-service/) for more information regarding the Agency Dashboard Service.

Basic Mechanism:

To use this document, you should be familiar with API concepts. All APIs return response as JSON by default. In case a response is need in XML the Content-Type of the request header needs to be set to either text/xml or application/xml.

This API will only work for the FunderID's for those Funder Dashboards that were created.

Data Provenance is Crossref.

Funder Agency Dashboard APIs:

Four APIs were created to return dashboard data.

<table>
<thead>
<tr>
<th>API Endpoint</th>
<th>HTTP Method</th>
<th>Description</th>
</tr>
</thead>
</table>

CHORUS – Funding Agency Dashboard Service API v1.1
<table>
<thead>
<tr>
<th>URL Path</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/agencies/:id/histories</td>
<td>GET</td>
<td>Returns data of indicator history table</td>
</tr>
<tr>
<td>/agencies/:id/publishers</td>
<td>GET</td>
<td>Returns data of publisher breakdown table</td>
</tr>
<tr>
<td>/agencies/:id/histories/yyyy/:mm/:dd</td>
<td>GET</td>
<td>Returns works associated with numbers displayed in indicator history table</td>
</tr>
<tr>
<td>/agencies/:id/histories/current</td>
<td>GET</td>
<td>Returns works associated with ALL numbers displayed in indicator history table</td>
</tr>
<tr>
<td>/agencies/:id/publishers/:memberId</td>
<td>GET</td>
<td>Returns work records corresponding to numbers in publisher table</td>
</tr>
</tbody>
</table>

**API 1: For fetching the data of indicator history table**

**API Endpoint:** /agencies/:id/histories

**Method:** GET

**Parameters to be passed:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:id(Required)</td>
<td>URL</td>
<td>Funder Id of agency whose data is to be fetched</td>
</tr>
</tbody>
</table>

**Headers:**
In order to request data in XML format, following header needs to be set for the API call:
'Content-Type: text/xml'
OR
'Content-Type: application/xml' (JSON is returned by default)

**Response Codes:**
200 – Successful Request  
404 – Requested agency not found in dashboard database  
500 – Something went wrong with the server

**Response:**
- agency_id – ID of agency/funder whose data was requested  
- agency_name – Name of Agency  
- tallies – An array of object corresponding to all rows in indicator history table

**Example cURL requests:**
curl "https://api.chorusaccess.org/v1.1/agencies/100000001/histories"
In case data is needed as XML:
curl -H "Content-Type: text/xml"
"https://api.chorusaccess.org/v1.1/agencies/100000001/histories"

Sample Response:
{
"agency_id": "100000001",
"agency_name": "National Science Foundation",
"tallies": [
{
"date": "12/31/2016",
"total_works": 97841,
"publicly_accessible_yes": 19599,
"publicly_accessible_unknown": 78272,
"reuse_terms_available_yes": 37056,
"reuse_terms_available_unknown": 60815,
"archived_yes": 82192,
"archived_unknown": 15679,
"passed_overall_audit": 16571
}
]
}

API 2: For fetching the data of publisher breakdown table
API Endpoint: /agencies/:id/publishers
Method: GET
Note(s): The parameters, headers, and response codes section for this API are the same as in API 1 above.

Response:
- agency_id – ID of agency/funder whose data was requested
- agency_name – Name of Agency
- publishers – An array of object corresponding to all rows in publisher table

Example cURL requests:
curl "https://api.chorusaccess.org/v1.1/agencies/100000001/publishers"
In case data is needed as XML:
curl -H "Content-Type: text/xml"
"https://api.chorusaccess.org/v1.1/agencies/100000001/publishers"

Sample Response:
{
"agency_id": "100000001",
"agency_name": "National Science Foundation",
"publishers": [
{
"publisher": "American Diabetes Association",
"total_works": 4,
"publicly_accessible_yes": 3,
"publicly_accessible_unknown": 1,
"reuse_terms_available_yes": 4,
"reuse_terms_available_unknown": 0,
"archived_yes": 4,
"archived_unknown": 0,
"passed_overall_audit": 3
}
}

API 3: For fetching the works associated with numbers displayed in indicator history table
API Endpoint: /agencies/:id/histories/:yyyy/:mm/:dd
Method: GET
Note(s): This API returns 50 records per request; setting limit query parameter to a higher value while making request can increase this limit. The headers and response codes section for this API are the same as in API 1 above.

Parameters to be passed:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:id (Required)</td>
<td>URL</td>
<td>Funder Id of agency whose data is to be fetched</td>
</tr>
<tr>
<td>:yyyy (Required)</td>
<td>URL</td>
<td>Year of date of which data is to be fetched</td>
</tr>
<tr>
<td>:mm (Required)</td>
<td>URL</td>
<td>Month of date of which data is to be fetched</td>
</tr>
<tr>
<td>:dd</td>
<td>URL</td>
<td>Day of date of which data is to be fetched</td>
</tr>
</tbody>
</table>
### (Required)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>category</td>
<td>Query</td>
<td>This can be one of, publicly accessible, archive, license and total acceptable. Default value for this parameter is ‘all’.</td>
</tr>
<tr>
<td>subcategory</td>
<td>Query</td>
<td>This can be one of 'yes' or 'unknown'. If only category is passed as query parameter with request, subcategory defaults to 'yes'</td>
</tr>
<tr>
<td>limit</td>
<td>Query</td>
<td>Number of records to be returned per request/page. Defaults to 50</td>
</tr>
<tr>
<td>offset</td>
<td>Query</td>
<td>Pagination offset. Default value is 0</td>
</tr>
</tbody>
</table>

### Optional

### Query

- This can be one of, publicly accessible, archive, license and total acceptable. Default value for this parameter is ‘all’.
- This can be one of 'yes' or 'unknown'. If only category is passed as query parameter with request, subcategory defaults to 'yes'.
- Number of records to be returned per request/page. Defaults to 50.
- Pagination offset. Default value is 0.

### Note:

- exchanging (:yyyy/:mm/:dd) with “current” will yield ALL results up to the current date.

### Response:

- agency_id – ID of agency/funder whose data was requested
- agency_name – Name of Agency
- breakdown_for – Date whose data is requested
- category – category parameter passed with request
- subcategory – subcategory parameter that was passed with request
- items – Array of work records
- limit – Number of records requested
- offset – Result offset
- total_results – Total number records matching the request

### Example cURL requests:

curl
"https://api.chorusaccess.org/v1.1/agencies/100000161(histories/2015/2/23?category=archive&subcategory=yes&limit=100"

In case data is needed as XML:
curl -H "Content-Type: text/xml"
"https://api.chorusaccess.org/v1.1/agencies/100000161(histories/2015/2/23?category=archive&subcategory=yes&limit=100"

### Sample Response:

```json
{
   "agency_id": "100000161",
   "agency_name": "National Institute of Standards and Technology",
}
```
"breakdown_for": "2015/2/23",
"category": "archive",
"subcategory": "yes",
"items": [
{
"DOI": "10.1002/2014jd022555",
"URL": "http://dx.doi.org/10.1002/2014jd022555",
"publisher": "Wiley-Blackwell",
"title": "Toward quantification and source sector identification of fossil fuel CO2 emissions from an urban area: Results from the INFLUX experiment",
"journal_name": "Journal of Geophysical Research: Atmospheres",
"last_audited_on": "1/12/2015",
"publicly_accessible_on_publisher_site": "1/12/2015",
"updated": "6/23/2017",
"added": "12/12/2014",
"published_print": "1/16/2015",
"published_online": "1/12/2015",
"license_type": "VOR",
"reuse_license_start_date": "01/12/2015",
"type": "journal-article",
"grant_ids": null,
"agency_portal_url": null
}]
"limit": 1,
"offset": 0,
"total_results": 61
}
API 4: For fetching work records corresponding to numbers in publisher table

**API Endpoint:** /agencies/:id/publishers/:memberId

**Method:** GET

**Note(s):** This API returns 50 records per request; setting limit query parameter to a higher value while making request can increase this limit. The headers and response codes section for this API are the same as in API 1 above.

### Parameters to be passed:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:id (Required)</td>
<td>URL</td>
<td>Funder ID of agency whose data is to be fetched</td>
</tr>
<tr>
<td>:memberId (Required)</td>
<td>URL</td>
<td>Member id of publisher whose data is to be fetched</td>
</tr>
<tr>
<td>category (Optional)</td>
<td>Query</td>
<td>This can be one of, publicly_accessible, archive, license and total_acceptable. Default value for this parameter is ‘all’.</td>
</tr>
<tr>
<td>subcategory (Optional)</td>
<td>Query</td>
<td>This can be one of 'yes' or 'unknown'. If only category is passed as query parameter with request, subcategory defaults to 'yes'</td>
</tr>
<tr>
<td>limit (Optional)</td>
<td>Query</td>
<td>Number of records to be returned per request/page. Defaults to 50</td>
</tr>
<tr>
<td>offset (Optional)</td>
<td>Query</td>
<td>Pagination offset. Default value is 0</td>
</tr>
</tbody>
</table>

**Response:**

- agency_id – ID of agency/funder whose data was requested
- agency_name – Name of Agency
- breakdown_for – Publisher whose data is requested
- category – category parameter passed with request
- subcategory – subcategory parameter that was passed with request
- items – Array of work records
- limit – Number of records requested
- offset – Result offset
- total_results – Total number records matching the request

**Example cURL requests:**
curl
"https://api.chorusaccess.org/v1.1/agencies/10000161/publishers/16?category=publicly_accessible&subcategory=yes"
In case data is needed as XML:
curl -H "Content-Type: text/xml"
"https://api.chorusaccess.org/v1.1/agencies/10000161/publishers/16?category=publicly_accessible&subcategory=yes"

Sample Response:
{
"agency_id": "10000161",
"agency_name": "National Institute of Standards and Technology",
"breakdown_for": "American Physical Society (APS)",
"category": "publicly_accessible",
"subcategory": "yes",
"items": [
{
"DOI": "10.1103/physreva.83.032501",
"URL": "http://dx.doi.org/10.1103/physreva.83.032501",
"publisher": "American Physical Society (APS)",
"authors": ["Pachucki Krzysztof", "Komasa Jacek"],
"title": "Magnetic dipole transitions in the hydrogen molecule",
"journal_name": "Physical Review A",
"last_audited_on": "11/1/2014",
"publicly_accessible_on_publisher_site": "3/1/2011",
"orcid_profile": [
{"family": "Komasa",
"given": "Jacek",
"ORCID": "http://orcid.org/0000-0002-9876-1304",
"_id": "56b246978a2395e2a5e822c5"
}
],
"updated": "6/23/2017",
"added": null,
"published_print": null,
"published_online": "3/1/2011",
"license_type": "VOR",
"reuse_license_start_date": "03/01/2011",
}
"type": "journal-article",
"agency_portal_url": null,
"grant_ids": null},
],
"limit": 1,
"offset": 0,
"total_results": 49
}

API Change Log:

<table>
<thead>
<tr>
<th>Revisions Date</th>
<th>Change</th>
<th>Reason for change</th>
</tr>
</thead>
<tbody>
<tr>
<td>6/20/2016</td>
<td>Added Data Provenance</td>
<td>Clarification</td>
</tr>
<tr>
<td>11/6/2016</td>
<td>Added additional parameters to API 3</td>
<td>To return results of data up to the current date, not limited to a particular date parameter</td>
</tr>
<tr>
<td>11/29/2016</td>
<td>• Changed and tags added to API 3&lt;br&gt;• issue_date changed to DOI_Issued_date&lt;br&gt;• publicly_accessible change to publicly_accessible_on_publisher_site&lt;br&gt;• verified changed to “Last Audited On”&lt;br&gt;• license changed to “Reuse License Start Date”&lt;br&gt;• added: &quot;License Type&quot;</td>
<td>Clarification</td>
</tr>
<tr>
<td>5/2017</td>
<td>Added the following fields&lt;br&gt;• published_online&lt;br&gt;• published_print&lt;br&gt;• added&lt;br&gt;• updated&lt;br&gt;• agency portal URL&lt;br&gt;• grant ids&lt;br&gt;• type</td>
<td>Align metadata found in dashboards and reports.</td>
</tr>
</tbody>
</table>

Change Notification Process:
Changes and/or updates will be communicated by notifying the technical representative noted in the CHORUS participant account details.

**Issues/questions:**

If you find a bug in this documentation or would like to propose an improvement to the API, please send an email to (support@chorusaccess.org). If you have a suggestion how to fix it, include that as well.