# Public Query Interface Version 1.0

## Committee Specification Draft 02 / Public Review Draft 01

## 13 October 2016

### Specification URIs

**This version:**

http://docs.oasis-open.org/coel/PQI/v1.0/csprd01/PQI-v1.0-csprd01.docx (Authoritative)
http://docs.oasis-open.org/coel/PQI/v1.0/csprd01/PQI-v1.0-csprd01.html
http://docs.oasis-open.org/coel/PQI/v1.0/csprd01/PQI-v1.0-csprd01.pdf

**Previous version:**

http://docs.oasis-open.org/coel/PQI/v1.0/csd01/PQI-v1.0-csd01.docx (Authoritative)
http://docs.oasis-open.org/coel/PQI/v1.0/csd01/PQI-v1.0-csd01.html
http://docs.oasis-open.org/coel/PQI/v1.0/csd01/PQI-v1.0-csd01.pdf

**Latest version:**

http://docs.oasis-open.org/coel/PQI/v1.0/PQI-v1.0.docx (Authoritative)
http://docs.oasis-open.org/coel/PQI/v1.0/PQI-v1.0.html
http://docs.oasis-open.org/coel/PQI/v1.0/PQI-v1.0.pdf

**Technical Committee:**

OASIS Classification of Everyday Living (COEL) TC

**Chairs:**

David Snelling (David.Snelling@UK.Fujitsu.com), Fujitsu Limited
Joss Langford (joss@activinsights.co.uk), Activinsights Ltd

**Editor:**

David Snelling (David.Snelling@UK.Fujitsu.com), Fujitsu Limited

**Related work:**

This specification is related to:

- *Classification of Everyday Living Version 1.0.* Edited by Joss Langford. Latest version: http://docs.oasis-open.org/coel/COEL/v1.0/COEL-v1.0.html.
- *Roles, Principles, and Ecosystem Version 1.0*. Edited by Matthew Reed. Latest version: http://docs.oasis-open.org/coel/RPE/v1.0/RPE-v1.0.html.
- *Behavioural Atom Protocol Version 1.0*. Edited by Joss Langford. Latest version: http://docs.oasis-open.org/coel/BAP/v1.0/BAP-v1.0.html.
- *Minimal Management Interface Version 1.0*. Edited by David Snelling. Latest version: http://docs.oasis-open.org/coel/MMI/v1.0/MMI-v1.0.html.
- *Identity Authority Interface Version 1.0*. Edited by Paul Bruton. Latest version: http://docs.oasis-open.org/coel/IDA/v1.0/IDA-v1.0.html.

**Abstract:**

This document describes the minimum synchronous query interface that will be provided by a Data Engine. Individual implementations of a Data Engine can provide further capabilities.

**Status:**

This document was last revised or approved by the OASIS Classification of Everyday Living (COEL) TC on the above date. The level of approval is also listed above. Check the “Latest version" location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Technical Committee (TC) are listed at https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=coel#technical.

TC members should send comments on this specification to the TC's email list. Others should send comments to the TC's public comment list, after subscribing to it by following the instructions at the "Send A Comment button on the TC's web page at https://www.oasis-open.org/committees/coel/.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the TC's web page (https://www.oasis-open.org/committees/coel/ipr.php).

### Citation format:

When referencing this specification the following citation format should be used:

**[COEL-PQI-v1.0]**

*Public Query Interface Version 1.0*. Edited by David Snelling. 13 October 2016. OASIS Committee Specification Draft 02 / Public Review Draft 01. http://docs.oasis-open.org/coel/PQI/v1.0/csprd01/PQI-v1.0-csprd01.html. Latest version: http://docs.oasis-open.org/coel/PQI/v1.0/PQI-v1.0.html.

# Notices

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see https://www.oasis-open.org/policies-guidelines/trademark for above guidance.

# Table of Contents

# 1 Introduction

This document describes the minimum synchronous query interface that MUST be provided by a Data Engine. Individual implementations of a Data Engine can provide further capabilities.

## 1.1 Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in **[RFC2119]**.

## 1.2 Normative References

**[RFC2119]** Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels”, BCP 14, RFC 2119, March 1997. http://www.ietf.org/rfc/rfc2119.txt.

**[RFC4627]** D. Crockford, The application/json Media Type for JavaScript Object Notation (JSON), July 2006, http://www.ietf.org/rfc/rfc4627.txt.

**[COEL_RPE-1.0]** *Roles, Principles, and Ecosystem Version 1.0.* Latest version: http://docs.oasis-open.org/coel/RPE/v1.0/RPE-v1.0.docx.

**[COEL_IDA-1.0]** *Identity Authority Interface Version 1.0.* Latest version: http://docs.oasis-open.org/coel/IDA/v1.0/IDA-v1.0.docx

**[COEL_BAP-1.0]** *Behavioural Atom Protocol Version 1.0.* Latest version: http://docs.oasis-open.org/coel/BAP/v1.0/BAP-v1.0.docx

**[ISO/IEC 5218]** Codes for the representation of human sexes, December 2004. http://www.iso.org/iso/catalogue_detail.htm?csnumber=36266

## 1.3 Non-Normative References

**[Coelition]** http://www.coelition.org

**[Data to Life]** Reed, M. & Langford, J. (2013). Data to Life. Coelition, London. ISBN 978-0957609402

# 2 Interface Specification

The Data Engine query interface SHALL have one method POST. The body of the request SHALL contain the query. The response to a successful query SHALL be a list of JSON Atoms that are the results of the query OR the result of an aggregation. The Data Engine segment interface SHALL have one method POST. The body of the request shall be a Consumer ID. The response to a successful query SHALL be the segment data for that consumer.

## 2.1 Authentication and Authorisation

To access the Query API, callers need API Credentials with two components:

- A userid to identify the caller.
- A password to authenticate the caller.

HTTP basic authentication SHALL be used to authenticate calls to the API. Passwords SHOULD be 64 bytes in length and MUST be supplied as an ASCII string. This MUST be prefixed with the userid followed by a colon to form the token passed in the HTTP Authorisation Header.

Example:

```
"9abf5386-2ac6-4e61-abc4-6b809a85d6cb:J1dOeWJJOkd3akhnSn4ma007M
DtUMVAxISgyOn9jI2U9NHNdRi4hfiw9c2I8PURcVltNMWQkamsrfGR4T24vKA=="
```

If the userid is unrecognized, or the wrong password is supplied a HTTP status code *401 Invalid username or password* SHALL be returned.

## 2.2 Query Operation

Initiate the query contained in the body of the request and return the result of the query.

| API | Description |
|---|---|
| POST query | Send a query to the Data Engine and wait for the response containing the result. |

### 2.2.1 Request

| Parameter Name | Description | Type |
|---|---|---|
| **ConsumerID** | Pseudonymous Key representing the requesting Consumer who is the subject of the query (REQUIRED). | **String**: Format defined in [COEL_IDA-1.0]. |

| Parameter Name | Description | Type |
|---|---|---|
| **OperatorID** | Pseudonymous Key representing the consumer's Operator (OPTIONAL). | **String**: Format defined in [COEL_IDA-1.0]. |
| **TimeWindow** | Represents the time window(s) for the query (OPTIONAL). | **Object:** Composed of StartTime, EndTime, and BlockBy. |
| **StartTime** | Start of time interval to be included in the query. Time in seconds since 1/1/1970 UTC (OPTIONAL). If absent, 1/1/1970 is assumed. Atoms will be included if their start time comes after this time. | **Integer:** Seconds since 1/1/1970 UTC. |
| **EndTime** | End of time interval to be included in the query. Time in seconds since 1/1/1970 UTC (OPTIONAL). If absent, infinity is assumed. Atom will be excluded if their start time comes after this time. | **Integer:** Seconds since 1/1/1970 UTC. |
| **BlockBy** | If present the number of seconds in each block returned (OPTIONAL). If absent all Atoms in the time window are returned as a single block or used in the aggregation computation. | **Integer:** Block length in seconds. |
| **Query** | The query for this request. (OPTIONAL) | **JSON Object:** Format defined in Section 2.2.1.1 |

**Media type:**

```
application/json, text/json
```

#### 2.2.1.1 Query Object

The query object has the following JSON structure.

- Query: (OPTIONAL)
  - Filter:

        - ColName: column name
        - Comparator: one of "=", ">", ">=", "<", "<=", "!="
        - Value: comparison value
    - AND (list of length > 0) (OPTIONAL)
        - Filter, AND, OR
    - OR (list of length > 0) (OPTIONAL)
        - Filter, AND, OR
    - NOT (OPTIONAL)
        - Filter, AND, OR
    - Aggregate (OPTIONAL)
        - Columns (list)
            - ColName: column name, see below
            - Aggregator: aggregator function, one of AVG, SUM, COUNT, MIN, MAX, STDDEV
        - GroupBy (list) (OPTIONAL)
            - ColName: column name
    - Project (OPTIONAL)
        - Include (list)
            - ColName: column name
        - Exclude (list)
            - ColName: column name

### 2.2.1.2 Column Names

The following table contains the column names that MUST be used in in queries and that the Data Engine has used to map the corresponding tag values from the Atoms posted.

| Name | Data type |
|---|---|
| HEADER_VERSION | [short, short, short, short] |
| WHEN_UTCOFFSET | int |
| WHEN_ACCURACY | int |
| WHEN_DURATION | int |
| WHAT_CLUSTER | short |
| WHAT_CLASS | short |
| WHAT_SUBCLASS | short |
| WHAT_ELEMENT | short |
| HOW_HOW | int |
| HOW_CERTAINTY | int |
| HOW_RELIABILITY | int |
| CONTEXT_SOCIAL | int |
| CONTEXT_WEATHER | int |
| CONTEXT_CONTEXTTAG | int |
| CONTEXT_CONTEXTVALUE | int |
| WHERE_EXACTNESS | int |
| WHERE_LATITUDE | double |

| | |
|---|---|
| WHERE_LONGITUDE | double |
| WHERE_W3W | string |
| WHERE_PLACE | int |
| WHERE_POSTCODE | string |
| CONSENT_ JURISDICTION | string |
| CONSENT_CONSENTDATE | int |
| CONSENT_RETENTIONPERIOD | int |
| CONSENT_PURPOSE | int |
| CONSENT_POLICYURL | string |
| CONSENT_WEBTOKENID | string |
| CONSENT_RECEIPTSERVICE | string |
| EXTENSION_INTTAG | int |
| EXTENSION_INTVALUE | int |
| EXTENSION_FLTTAG | int |
| EXTENSION_FLTVALUE | double |
| EXTENSION_STRTAG | int |
| EXTENSION_STRVALUE | string |

## 2.2.2 Response

There are three possible responses to a query. If successful and the Data Engine choses to return the query result immediately, an HTTP status code of 200 *OK* MUST be returned and the QueryResult element included in the body of the response. The Data Engine MAY chose to create a separate resource where the client can obtain the query result, if for example the query response is very large. In this case the Data Engine MUST return an HTTP status code 201 *Created* and set the “Location:” header to the URL where the QueryResult can be obtained with (a possibly paged) GET request. In this case the response MAY include the ResultCreated element. Lastly, if unsuccessful, an HTTP error code SHOULD be returned and a JSON object MAY be returned providing some explanation of the failure.

| **Parameter Name** | **Description** | **Type** |
|---|---|---|
| **QueryResult** | The query result is a list of JSON objects that match the query. | **JSON Object:** Format defined in Section 2.2.2.1 |
| **ResultCreated** | This element describes the query result’s size, availability, and location. | **JSON Object:** Format defined in Section 2.2.2.2 |

| Parameter Name | Description | Type |
|---|---|---|
| **Reason** | In case of a failure, this is an optional description of why the query failed. | **String:** |

**Media type:**

```
application/json, text/json
```

### 2.2.2.1 QueryResult Object

For a simple filter the result is a JSON list of Atoms, see [COEL_BAP-1.0]. If a projection is specified only requested fields of the matching Atoms are included.

For aggregates, the result objects contain a list of aggregated columns, described by column name and aggregator (as specified in the query), with the result of the aggregate function. If a grouping is specified the object contains a list of column names and their groups for each aggregation.

When BlockBy is absent, all results are returned as the only element in the Blocks list.

Blocks: (list)

- Aggregate: (list)
  - ColName: column name
  - Aggregator: aggregate function
  - Value: aggregate function value
- Group: (list)
  - ColName: grouping column
  - Value: group

### 2.2.2.2 ResultCreated Object

The ResultCreated element is a JSON object with the following fields:

| Name | Value | Description | REQUIRED |
|---|---|---|---|
| Size | Integer | The expected size in bytes of the QueryResult object. | No |
| Location | String | The location (MUST be the same as in the Location: header) where the QueryResult can be obtained. | No |
| AvailableFrom | Integer | Time from which the QueryResult can be obtained, presented in seconds since 1970/01/01 00:00Z (Unix timestamp in UTC) | No |
| AvailableUntil | Integer | Time until which the QueryResult can be obtained, presented in seconds since 1970/01/01 00:00Z (Unix timestamp in UTC) | No |

## 2.3 Segment Data

Request segment data for a Consumer.

| API | Description |
|---|---|
| POST segment | Retrieve a copy of all available segment data for the given consumer. |

### 2.3.1 Request

| Parameter Name | Description | Type |
|---|---|---|
| **ConsumerID** | Pseudonymous Key representing the requesting Consumer who is the subject of the query (REQUIRED). | **String**: Format defined in [COEL_IDA-1.0]. |

**Media type:**

```
application/json, text/json
```

### 2.3.2 Response

If successful, an HTTP status code of 200 *OK* MUST be returned along with the Segment Data. If unsuccessful, an HTTP error code SHOULD be returned, in which case a JSON object MAY be returned providing some explanation of the failure, see section 2.2.2. The Gender parameter SHALL have enumerated fields reserved for compliance with **[ISO/IEC 5218]**.

| Parameter Name | Description | Type |
|---|---|---|
| **SegmentData** | An OPTIONAL object containing (OPTIONALLY) residential time zone and latitude, gender, and year of birth. | **Object:** Composed of ResidentTimeZone, ResidentLatitude, Gender, and YearOfBirth. |

| Parameter Name | Description | Type |
|---|---|---|
| **ResidentTimeZone** | The time zone in which the Consumer generally resides. | **TimeZoneString:** As +/-hh:mm from UTC. |
| **ResidentLatitude** | The latitude (rounded to an integer) at which the Consumer generally resides. | **Integer:** Representing latitude rounded to an integer. |
| **Gender** | The gender of the Consumer. | **Integer 0-99:**<br>0 not known<br>1 male<br>2 female<br>9 not applicable |
| **YearOfBirth** | Year in which the Consumer was born. | **Integer:** Representing year of birth. |
| **Reason** | In case of a failure, this is an optional description of why the query failed. | **String:** |

**Media type:**

```
application/json, text/json
```

**Sample:**

```
{"SegmentData":
      {"ResidentTimeZone": "+03:00",
       "ResidentLatitude": 51,
       "Gender": 2,
       "YearOfBirth": 1993
      }
}
```

# 3 Conformance

Any implementation MUST accept queries in the form described in section 2 of this document AND the conformance criteria in **[COEL_RPE-1.0]**, however only a minimum functionality MUST be supported.

- A Data Engine MUST return raw atoms within a time window for a given ConsumerID.
- A Data Engine MUST return the number of atoms held in a time window for a given ConsumerID.

The following is the first of the two minimum queries that a Data Engine implementation MUST support. The result of this query is a list of all Atoms with a start time within the time window.

**Sample:**

```
{"ConsumerID" : "ed58fc40-a866-11e4-bcd8-0800200c9a66",
 "Timewindow" : {
    "StartTime" : 1415145600,
    "EndTime" : 1415232000
   }
}
```

The following is the second of the two minimum queries that a Data Engine implementation MUST support. The result of this query is the number of Atoms with a start time within the time window.

**Sample:**

```
{"ConsumerID" : "ed58fc40-a866-11e4-bcd8-0800200c9a66",
 "Timewindow" : {
    "StartTime" : 1415145600,
    "EndTime" : 1415232000},
 "Query" : {
    "Aggregate" : {
        "Columns" : {
            "ColName" : "WHAT_CLUSTER",
            "Aggregator" : "COUNT"}}}
}
```

Any implementation MUST implement the segment query as described in section 2.3.

Any implementation MUST implement either the 200 *OK* or the 201 *Created* response pattern and MAY implement both. See section 2.2.2.

# Appendix A. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

**Participants:**

Paul Bruton, Individual Member
Joss Langford, Activinsights
Matthew Reed, Coelition
David Snelling, Fujitsu

# Appendix B. Revision History

| Revision | Date | Editor | Changes Made |
|---|---|---|---|
| 1 | 22/09/2105 | David Snelling | Initial inclusion of submission material. |
| 2 | 25/09/2105 | Paul Bruton | Added review comments |
| 3 | 25/09/2105 | Joss Langford | Review, spelling correct. |
| 4 | 11/10/2015 | David Snelling | Reviewed in document comments and fixed or created issues. Fixed issue: COEL-9 |
| 5 | 11/10/2015 | David Snelling | Removed tracking |
| 6 | 11/10/2015 | David Snelling | Added column names table. |
| 7 | 13/10/2015 | Paul Bruton | Conformance includes reference to RPE document. |
| 8 | 19/10/2015 | David Snelling | Fixed silly quotes and general tidy up. |
| 9 | 31/10/2015 | Joss Langford | Accept all changes, track changes off, check references and style consistency. |
| 10 | 02/11/2015 | David Snelling | Final Data Change |
| 11 | 03/11/2015 | Paul Bruton | Added normative terms in 1st paragraph of section 2, corrected text in description of password encoding |
| 12 | 24/11/2015 | Paul Bruton | Addressing issues COEL-43 and COEL-44 |
| 13 | 25/11/2015 | David Snelling | Set date for CD publication |
| 14 | 07/01/2016 | David Snelling | Update to WD02 and changed error code management in line with issue COEL-42. |
| 15 | 14/01/2016 | Paul Bruton | Reviewed error codes and added ‘Reason’ field to response in 2.3.2 |
| 16 | 12/02/2016 | Paul Bruton | Accepted all previous edits |
| 17 | 29/06/2016 | Dave Snelling | Included created resource pattern for large query results, COEL-12. |
| 18 | 01/07/2016 | Dave Snelling | Fixed a few typos and remover change tracking. |
| 19 | 21/08/2016 | Joss Langford | Gender field of segment data updated (COEL-74). |
| 20 | 26/08/2016 | Joss Langford | Gender example fixed. |
| 21 | 31/08/2016 | David Snelling | Fixed Location and Version Column Names, COEL-78 |
| 22 | 31/08/2016 | David Snelling | Added Consent terms to Column Names, COEL-78 |
| 23 | 31/08/2016 | David Snelling | War with Word autoformatting. |

|  |  |  |  |
|---|---|---|---|
| 24 | 16/09/2016 | Joss Langford | Reference correction COEL-81 |
| 25 | 23/09/2016 | Paul Bruton | Segment method referenced in preamble and conformance sections. Slight change to wording in segment to clarify that data are being retrieved from data engine. |
| 26 | 10/10/2016 | Joss Langford | Changes accepted |