

Saved Queries API

In apstrata, [Documents](#) that are saved in a [Store](#) can be retrieved by running a *Dynamic Query* or a *Saved Query*.

A *Dynamic Query* is one that is built every time on the fly by specifying the condition, fields to return, sort order, and aggregates.

A *Saved Query* is one whose definition is saved for later use. Rather than every time specifying the condition, fields to return, sort order, and aggregates, a developer can add a *Saved Query* whose definition specifies all the previously mentioned query constructs. Each *Saved Query* has a name which is used to run the *Query* and get back the results.

Moreover, a developer can implement a pagination model for the matching results by specifying the number of *Documents* per page as well as the number of the page to return. This is very useful for cases where a huge number of [Documents](#) matches the *Query* condition. In these cases, the results will be returned in pages or chunks rather than a whole block which impacts the performance of the application at hand.

Furthermore, when running a *Query*, a developer can look for documents matching a specific full text search criteria. In this case, documents satisfying both the *Query* condition and the full text search criteria will be returned in the results.

In terms of security, *Documents* or fields within a *Document* are returned in the results only if their read [ACL](#) contains the [User](#) or one of the [Groups](#) of the [User](#) running the *Query*.

A developer may run as many *Queries* as needed per store. However, based on the apstrata account type, there is a limit on the number of predicates used in the *Query* condition, on the number of *Documents* returned in the results, and on the total size of the result set.

Saved Query XML Schema Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="query">
    <xs:annotation>
      <xs:documentation>Save Query root element</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="executeACL" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Specifies which user or group can execute the saved query</xs:documentation>
          </xs:annotation>
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:minLength value="1"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="store" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Specifies to which store the saved query applies</xs:documentation>
          </xs:annotation>
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:minLength value="1"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="condition" minOccurs="0">
          <xs:annotation>
            <xs:documentation>The saved query condition in apstrata syntax</xs:documentation>
          </xs:annotation>
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:minLength value="1"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="ftsQuery" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Set in case the user wants to run a full text search</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

```

</xs:annotation>
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="returnedFields" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Specifies which fields should be returned in result</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="field" maxOccurs="unbounded">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:minLength value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="sort" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Specifies on which field the sorting should be done, and the type and
order of the sorting(ascending or descending) </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="field" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="name">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:minLength value="1"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="type">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:minLength value="1"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="order">
              <xs:annotation>
                <xs:documentation>ASC(ascending) or DESC(descending)</xs:
documentation>
              </xs:annotation>
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:minLength value="1"/>
                  <xs:enumeration value="ASC"/>
                  <xs:enumeration value="DESC"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="count" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Specifies the number of records that match the query</xs:
documentation>
  </xs:annotation>

```

```

    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string"/>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="aggregate" minOccurs="0">
    <xs:annotation>
      <xs:documentation>Allows the user to execute an aggregate function on the results
returned by his query request</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="expression">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:minLength value="1"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="page" type="xs:boolean" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Defines the page number of the page to return in case
there is more than one</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="global" type="xs:boolean" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Specifies if the aggregate function should be executed on
all the documents returned by the condition</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="runAs" minOccurs="0">
    <xs:annotation>
      <xs:documentation>It allows the owner to run a service as one of his own users</xs:
documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="user">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:minLength value="1"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="deniedFields" type="xs:boolean" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Gives the owner the possibility to request the denied
fields for the runAs user</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="resultsPerPage" type="xs:positiveInteger" minOccurs="0">
    <xs:annotation>
      <xs:documentation>Determines the number of results per page to return</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="forceCurrentSnapshot" type="xs:boolean" minOccurs="0">
    <xs:annotation>
      <xs:documentation>In order to avoid data inconsistency, even for a short duration of
time,
the user can send this parameter with value "true" to query data
from the master database server.
Will query from slave database server if the value is set to
"false" </xs:documentation>

```

```
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

Saved Query Example

```
<query>
  <executeACL>creator</executeACL>
  <store>testStore</store>
  <condition><![CDATA[ startingDate<date>={startdate}]]></condition>
  <returnedFields>
    <field>firstName</field>
    <field>lastName </field>
  </returnedFields>
  <count>true</count>
  <resultsPerPage>10</resultsPerPage>
</query>
```