UI5

SAP UI5 is a framework for developing user interfaces based on cross-browser JavaScript library

Key features

- Documented API
- Extensible components
- High performance
- Flexible design options (theme designer, direct CSS editing)
- Ability to use JAX (in the future WS)
- Use jQuery library

SAP UI5 and OpenUI5

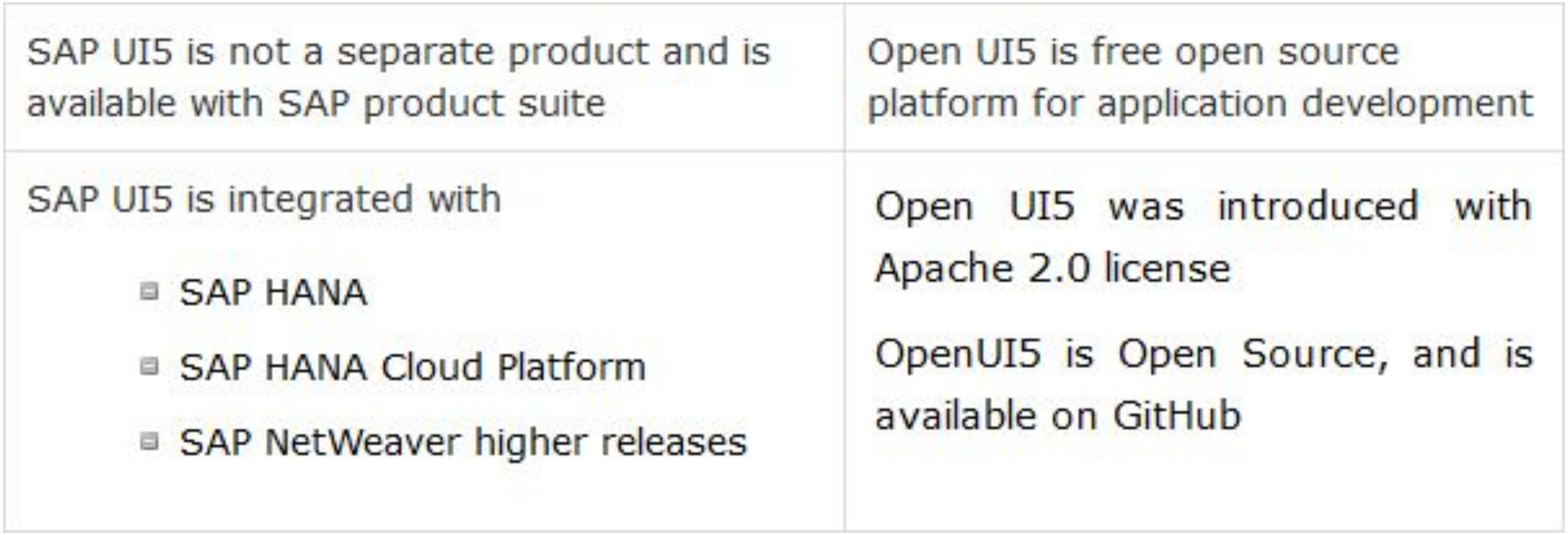
SAP UI5 and Open UI5, both provide the UI development environment.

They are different from each other in the following aspects − SAP UI5 is part of SAP product suite and is not a separate license. It is integrated with different SAP products like:

- **SAP NW 7.4 or higher**
- **SAP NetWeaver AS 7.3x**
- **SAP HANA Cloud and on premise solution**

Open UI5 is an open source technology for application development and it was released with Apache 2.0.

Main difference of OpenUI5 and SAP UI5



SAP UI5 is not a separate product and is available with SAP product suite	Open UI5 is free open source platform for application development
SAP UI5 is integrated with ▪ SAP HANA ▪ SAP HANA Cloud Platform ▪ SAP NetWeaver higher releases	Open UI5 was introduced with Apache 2.0 license OpenUI5 is Open Source, and is available on GitHub

UI5 Browser Support

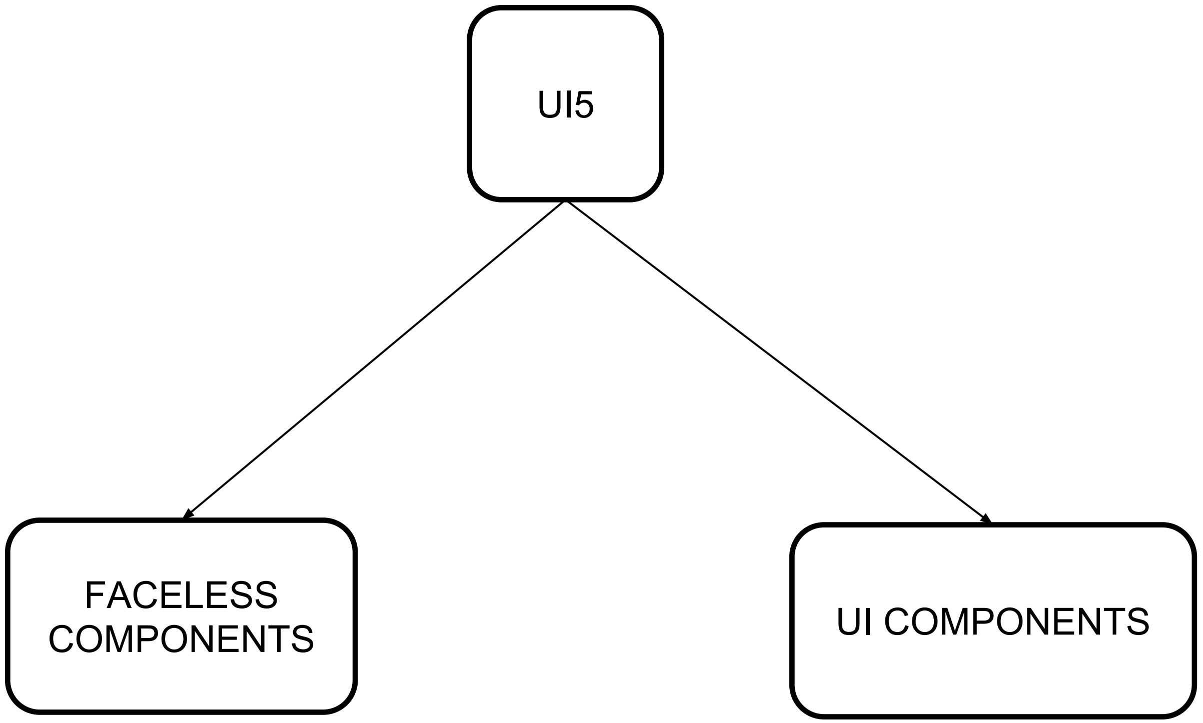
Firefox version 10 or higher

Google Chrome version 1 or higher

Safari browser version 3 or higher

Internet Explorer v8 or higher. IE 8 supports limited features of CSS3 like text shadows, and corners, etc.

UI5 components



UI Components

Client side component: This includes,

- Control libraries sap.m, sap.ui.common, etc.
- Core Javascript
- Test includes HTML and Javascript

Server side component

- Theming Generator
- Control and application development tools in Eclipse
- Resource handler

UI Components

Each component should contain the following files:

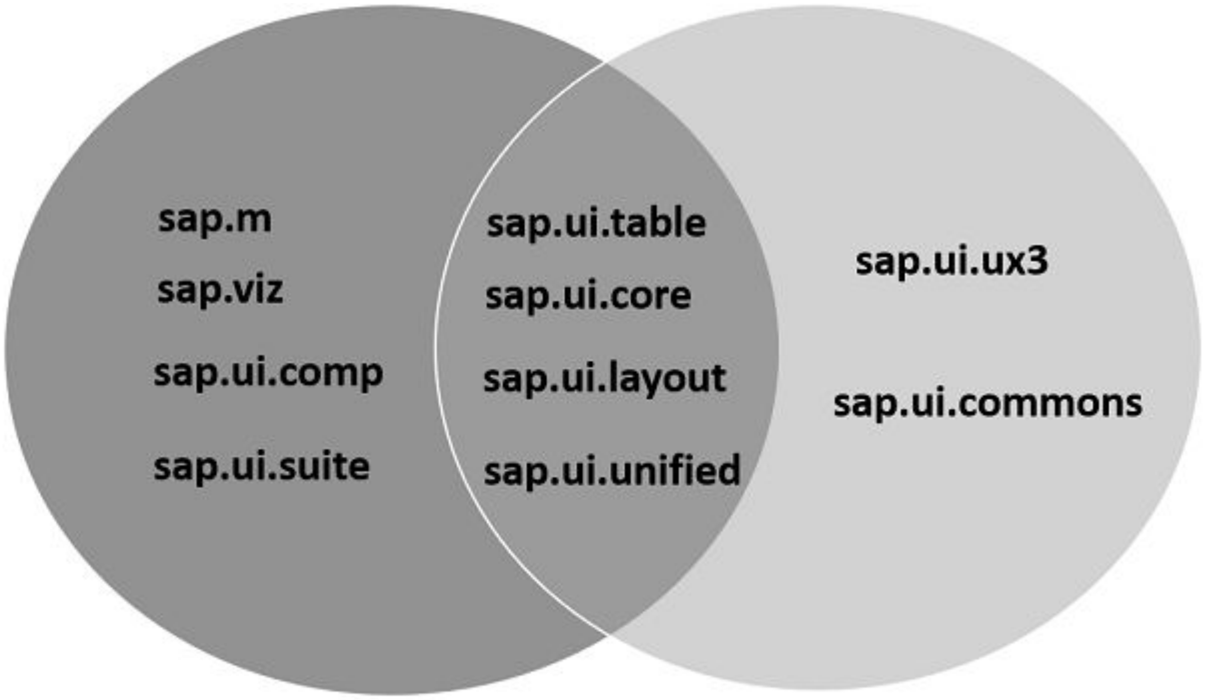
- **Component.json - file that contains metadata for design time and is used only for design time tools.**

- **Component.js – file, which used to define properties, events, and components methods that are responsible for runtime metadata.**

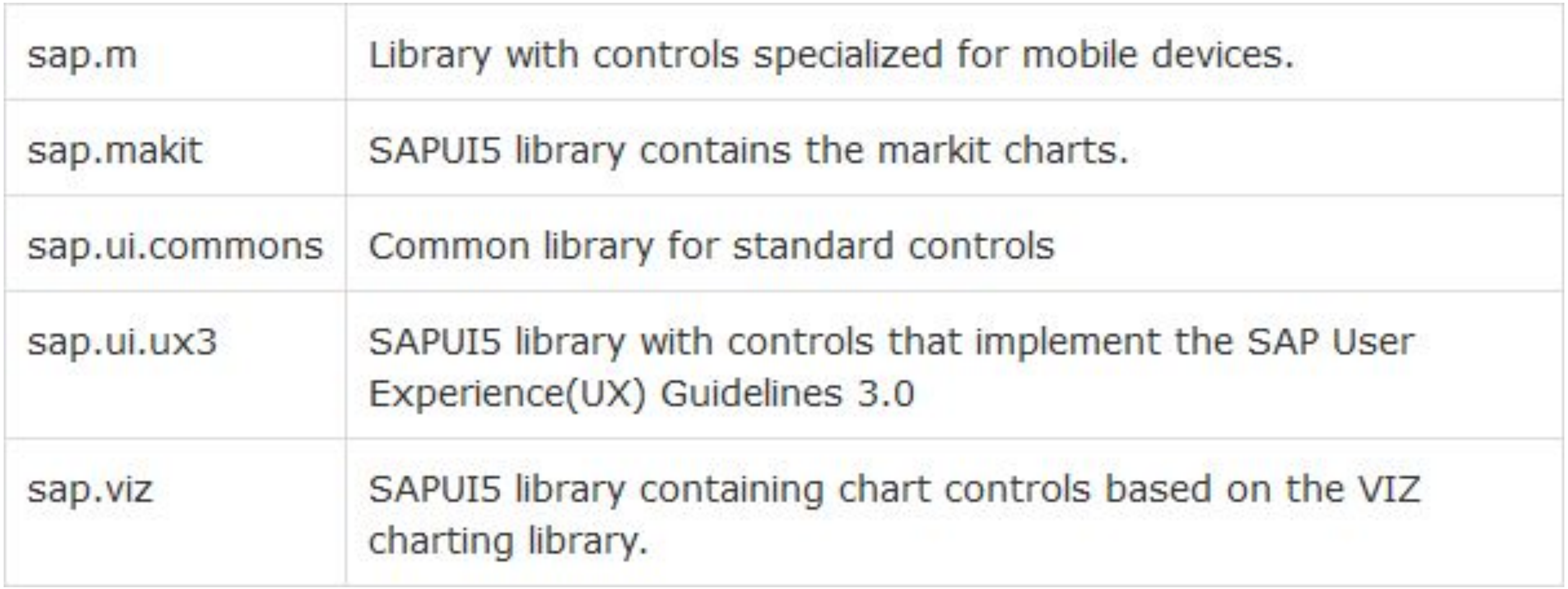
UI5 Control Libraries

Common SAPUI5 control libraries

- **Sap.ui.commons for control fields, buttons, etc.**
- **Sap.m is the most common control library and is used for mobile devices**
- **Sap.ui.table includes table control**
- **Sap.ui.ux3**



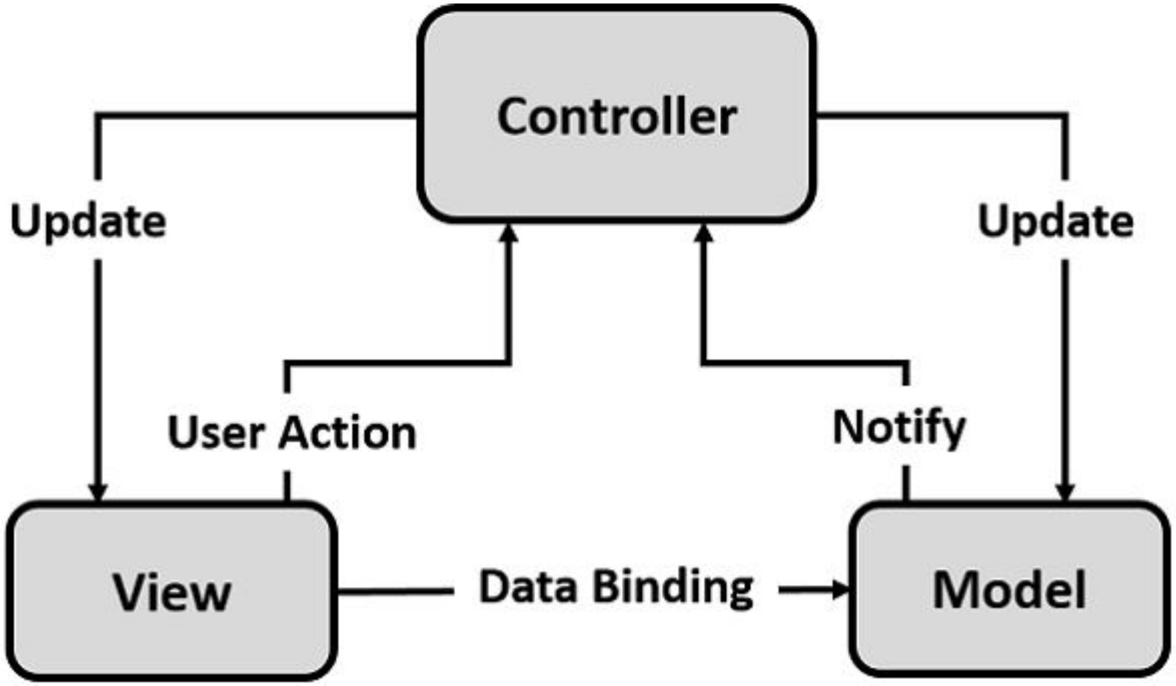
UI5 Control Libraries



sap.m	Library with controls specialized for mobile devices.
sap.makit	SAPUI5 library contains the markit charts.
sap.ui.commons	Common library for standard controls
sap.ui.ux3	SAPUI5 library with controls that implement the SAP User Experience(UX) Guidelines 3.0
sap.viz	SAPUI5 library containing chart controls based on the VIZ charting library.

MVC on UI5

- The Model is responsible for managing the application data in the database/backend.
- The View is responsible for defining the user interface to users. When a user sends a requests from his device, the view is responsible for data view as per the request submitted.
- The Controller is used to control the data and view events as per user interaction by updating the view and model.



UI5 Models

JSON Model

- JSON model is a client-side model and is used for small data sets.
- JSON model supports two-way binding. Data binding concept is mentioned in the latter half of this tutorial.
- JSON model can be used to bind controls to JavaScript object data.

XML Model

- XML model can be used to bind controls to XML data.
- XML is also a client side model and hence is used only for small data sets.
- XML model doesn’t provide any mechanism for server-based paging or loading of deltas.
- XML model also supports two-way data binding.

UI5 Views

Views are defined using SAP libraries as follows:

- **XML with HTML, mixed, or Standalone: Library- sap.ui.core.mvc.XMLView**
- **JavaScript: Library- sap.ui.core.mvc.JSView**
- **JSON: Library - sap.ui.core.mvc.JSONView**
- **HTML: Library - sap.ui.core.mvc.HTMLView**

JavaScript View Sample

```
Sap.ui.jsview(“sap.hcm.address”, {
  getControllerName: function() {
    return “sap.hcm.address”;
  },
  createContent: function(oController) {
    var oButton = new sap.ui.commons.Button({ text: “Hello” });
    oButton.attachPress(function() {
      oController.Hello();
    })
    Return oButton;
  }
});
```

HTML View Sample

```
<template data-controller-name = ”sap.hcm.address’>
  <h1>title</h1>
  <div> Embedded html </div>
  <div class = ”test” data-sap-ui-type = ”sap.ui.commons.Button”
    Id = ”Button1” data-text =  ”Hello” Data-press = ”sayHello”>
  </div>
</template>
```

JSON View Sample

```
{
   “type”:”sap.ui.core.mvc.JsonView”,
   “controllerName”:”sap.hcm.address”,
   ...........................
   .......................
   ........................
}
```

Comparison of View Types

Feature	JS View	XML View	JSON View	HTML View
Standard and Custom Libraries	Yes	Yes	Yes	Yes
Properties of types string, int Boolean, float	Yes	Yes	Yes	Yes
Aggregation 1:1, 1:n Association 1:1, 1:n	Yes	Yes	Yes	Yes
Simple Data Binding	Yes	Yes	Yes	Yes
Customize Data Binding	Yes	No	No	No
Embedded HTML	No	Yes	No	No
Code Completion	Yes	Yes	No	No
Templating	Yes	No	No	No
Validation	No	Yes	No	No
Single Event Listener	Yes	Yes	Yes	Yes

Start work with UI5

- **First step of work with UI5 is bootstrapping.**

```
<script id="sap-ui-bootstrap"
    type="text/javascript"
    src="resources/sap-ui-core.js"
    data-sap-ui-theme="sap_belize"
    data-sap-ui-libs="sap.ui.commons"
    data-sap-ui-compatVersion="edge">
</script>
```

Main types of UI elements

- **Simple Controls**

  Text fields, Buttons, Links, Pictures, etc.
- **Value Holder**

  Various selectors to select values, such as: lists, drop-down lists, selection buttons, etc.
- **Layouts**

  Matrix layers, Shapes, Separators, etc.
- **Complex Controls**

  Complex elements such as Menus, Tables, Hierarchical tables, etc.
- **Dialogs**

  Dialog, Message
- **UX3 Controls**

  UX3 components such as Shell, Facet filters, Tooltips, etc.
- **Others**

  All other elements

UI5 Core

- **Sap.ui.getCore() – method, which return a core instance.**
- **Sap.ui.getCore().byid(id) – method, which return an instance of UI5 control created with id.**
- **Sap.ui.getCore().applyChanges() – method, which carry out and render the changes for UI5 controls immediately.**
- **jQuery.sap.domById(id) - method, which return any HTML element with id. If there is a UI5 control with id, the element returned is top most HTML element of UI5 control.**
- **jQuery.sap.byId(id) – method, which return jQuery object of DOM element with specified Id.**

oData

- **This protocol enables applications to expose data, by using common Web technologies, and by means of a data service that can be consumed by clients within corporate networks and across the Internet.**

oData request syntax

Sample:

- **http://host/service/Customers?$expand=Orders&$filter=substringof(CompanyName, 'bikes')&$orderby=CompanyName asc&$top=2&$skip=3&$skiptoken='Contoso','AKFNU'&$inlinecount=allpages&$select=CustomerID,CustomerName,Orders**

$count

The $count system query option allows clients to request a count of the matching resources included with the resources in the response.

For example:

https://wus-odata.hawkeyecloud.com/v4/usage/v1/233ceacf-87f0-4d7c-80ec-17d118626916/3f149a6e-78ff-e511-80c8-000d3a32faa6/InteractionSummary/$count

Returns a count:

113

$expand

The $expand system query option specifies the related resources to be included in line with retrieved resources. You might think of this is an SQL join operation. Only navigation properties can be expanded; you can identify navigation propertes in the Data Model entity relationship diagram.

For example:

https://wus-odata.hawkeyecloud.com/v4/usage/v1/233ceacf-87f0-4d7c-80ec-17d118626916/3f149a6e-78ff-e511-80c8-000d3a32faa6/WorkflowActionSummary?$expand='WorkflowId'

$orderby

The $orderby system query option allows clients to request resources in a particular order.

For example:

https://wus-odata.hawkeyecloud.com/v4/usage/v1/233ceacf-87f0-4d7c-80ec-17d118626916/37df5e9d-5734-e611-80c3-000d3a320ed2/WorkflowInstanceActionByYear?$orderby=MaxDurationInsSeconds.

$search

The $search system query option allows clients to request entities matching a free-text search expression.

For example:

http://host/service/Products?$search=blue OR green

$select

The $select system query option allows clients to requests a specific set of properties for each entity or complex type.

For example:

http://host/service/Products?$select=Rating,ReleaseDate

$skip

The $top system query option requests the number of items in the queried collection to be included in the result. The $skip query option requests the number of items in the queried collection that are to be skipped and not included in the result. A client can request a particular page of items by combining $top and $skip.

For example:

https://wus-odata.hawkeyecloud.com/v4/usage/v1/233ceacf-87f0-4d7c-80ec-17d118626916/3f149a6e-78ff-e511-80c8-000d3a32faa6/WorkflowActionSummary?$skip=10

$top

The $top system query option requests the number of items in the queried collection to be included in the result. The $skip query option requests the number of items in the queried collection that are to be skipped and not included in the result. A client can request a particular page of items by combining $top and $skip.

For example:

https://wus-odata.hawkeyecloud.com/v4/usage/v1/233ceacf-87f0-4d7c-80ec-17d118626916/3f149a6e-78ff-e511-80c8-000d3a32faa6/WorkflowActionSummary?$top=10

$filter

The $filter system query option allows clients to filter a collection of resources that are addressed by a request URL.

Equal	filter=ActionCount+eq+10	Query on ActionCount to find ActionCount that equals 10.
Select a range of values	filter=Entry_No gt 610 and Entry_No lt 615	Query on Entry service. Returns entry numbers 611 through 614.
And	filter=Country_Region_Code eq 'ES' and Payment_Terms_Code eq '14 DAYS'	Query on Customer service. Returns customers in Spain where Payment_Terms_Code=14 DAYS.
Or	filter= Country_Region_Code eq 'ES' or Country_Region_Code eq 'US'	Query on Customer service. Returns customers in Spain and the United States. You can use OR operators to apply different filters on the same field. However, you cannot use OR operators to apply filters on two different fields.
Less than	filter=Entry_No lt 610	Query on GLEntry service. Returns entry numbers that are less than 610.
Greater than	filter= Entry_No gt 610	Query on GLEntry service. Returns entry numbers 611 and higher.
Greater than or equal to	filter=Entry_No ge 610	Query on GLEntry service. Returns entry numbers 610 and higher.

$filter

Less than or equal to	filter=Entry_No le 610	Query on GLEntry service. Returns entry numbers up to and including 610.
Different from (not equal)	filter=VAT_Bus_Posting_Group ne 'EXPORT'	Query on Customer service. Returns all customers with VAT_Bus_Posting_Group not equal to EXPORT.
endswith	filter=endswith(VAT_Bus_Posting_Group,'RT')	Query on Customer service. Returns all customers with VAT_Bus_Posting_Group values that end in RT.
startswith	filter=startswith(Name, 'S')	Query on Customer service. Returns all customers names beginning with “S”.
substringof	filter=substringof(Name, ‘urn’)	Query on Customer service. Returns customer records for customers with names containing the string “urn”.
length	filter=length(Name) gt 20	Query on Customer service. Returns customer records for customers with names longer than 20 characters.
indexof	filter=indexof(Location_Code, ‘BLUE’) eq 0	Query on Customer service. Returns customer records for customers having a location code beginning with the string BLUE.
replace	filter=replace(City, 'Miami', 'Tampa') eq 'CODERED'	
substring	filter=substring(Location_Code, 5) eq 'RED'	Query on Customer service. Returns true for customers with the string RED in their location code starting as position 5.

OData query cheat sheet

https://help.nintex.com/en-US/insight/OData/HE_CON_ODATAQueryCheatSheet.htm