

# AtScale OpenAPI

Below is the OpenAPI file for the AtScale public API.

```
{
  "openapi": "3.0.0",
  "paths": {
    "/v1/public/catalogs/{catalogId}/models/{modelId}/aggregates-creation/scheduler": {
      "get": {
        "operationId": "getCreationSchedules",
        "summary": "Get creation schedules for a model.",
        "parameters": [
          {
            "name": "Authorization",
            "required": false,
            "in": "header",
            "schema": {
              "type": "string"
            }
          },
          {
            "name": "catalogId",
            "required": true,
            "in": "path",
            "description": "The Catalog ID for which you want to list schedules.",
            "schema": {
              "type": "string"
            }
          },
          {
            "name": "modelId",
            "required": true,
            "in": "path",
            "description": "The Model ID for which you want to list schedules.",
            "schema": {
              "type": "string"
            }
          }
        ],
        "responses": {
          "200": {
            "description": "Success.",
            "content": {
              "application/json": {
                "schema": {
                  "type": "array",
                  "items": {
                    "$ref": "#/components/schemas/AggregateCreationScheduleResponseDto"
                  }
                }
              }
            }
          },
          "400": {
            "description": "Invalid request for example: Invalid parameters, missing parameters."
          },
          "401": {
            "description": "Unauthorized. This error will occur if calling user is not authenticated."
          },
          "403": {
```

```

    "description": "Forbidden. This error will occur if calling user does not have access right for the API."
  },
  "404": {
    "description": "Not found. The server cannot find the requested resource."
  },
  "500": {
    "description": "A server error occurred. Check the logs for more details."
  }
},
"tags": [
  "Aggregate Creation Schedule"
],
"security": [
  {
    "bearer": []
  }
]
},
"post": {
  "operationId": "createSchedule",
  "summary": "Create a new schedule using cron time expression.",
  "parameters": [
    {
      "name": "Authorization",
      "required": false,
      "in": "header",
      "schema": {
        "type": "string"
      }
    },
    {
      "name": "catalogId",
      "required": true,
      "in": "path",
      "description": "The Catalog ID for which you want to create a schedule.",
      "schema": {
        "type": "string"
      }
    },
    {
      "name": "modelId",
      "required": true,
      "in": "path",
      "description": "The Model ID for which you want to create a schedule.",
      "schema": {
        "type": "string"
      }
    }
  ],
  "requestBody": {
    "required": true,
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/AggregateCreationScheduleRequestDto"
        },
        "examples": {
          "example": {
            "value": {
              "timeExpression": "0 25 06 ? * *"
            }
          }
        }
      }
    }
  }
}

```

```

    }
  },
  "responses": {
    "201": {
      "description": "Success.",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/AggregateCreationScheduleResponseDto"
          }
        }
      }
    },
    "400": {
      "description": "Invalid request for example: Invalid parameters, missing parameters."
    },
    "401": {
      "description": "Unauthorized. This error will occur if calling user is not authenticated."
    },
    "403": {
      "description": "Forbidden. This error will occur if calling user does not have access right for the API."
    },
    "404": {
      "description": "Not found. The server cannot find the requested resource."
    },
    "500": {
      "description": "A server error occurred. Check the logs for more details."
    }
  },
  "tags": [
    "Aggregate Creation Schedule"
  ],
  "security": [
    {
      "bearer": []
    }
  ]
},
"/v1/public/catalogs/{catalogId}/models/{modelId}/aggregates-creation/scheduler/history/jobs/{jobId}": {
  "get": {
    "operationId": "getScheduleHistory",
    "summary": "Get the history of the schedule with the specified ID.",
    "parameters": [
      {
        "name": "Authorization",
        "required": false,
        "in": "header",
        "schema": {
          "type": "string"
        }
      },
      {
        "name": "catalogId",
        "required": true,
        "in": "path",
        "description": "The Catalog ID for which you want to get schedule history.",
        "schema": {
          "type": "string"
        }
      },
      {
        "name": "modelId",
        "required": true,

```

```

    "in": "path",
    "description": "The Model ID for which you want to get schedule history.",
    "schema": {
      "type": "string"
    }
  },
  {
    "name": "jobId",
    "required": true,
    "in": "path",
    "description": "The Job ID for which you want to get schedule history.",
    "schema": {
      "type": "string"
    }
  }
],
"responses": {
  "200": {
    "description": "Success.",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/AggregateScheduleHistoryResponseDto"
        }
      }
    }
  },
  "400": {
    "description": "Invalid request for example: Invalid parameters, missing parameters."
  },
  "401": {
    "description": "Unauthorized. This error will occur if calling user is not authenticated."
  },
  "403": {
    "description": "Forbidden. This error will occur if calling user does not have access right for the API."
  },
  "404": {
    "description": "Not found. The server cannot find the requested resource."
  },
  "500": {
    "description": "A server error occurred. Check the logs for more details."
  }
},
"tags": [
  "Aggregate Creation Schedule"
],
"security": [
  {
    "bearer": []
  }
]
},
"/v1/public/catalogs/{catalogId}/models/{modelId}/aggregates-creation/scheduler/jobs/{jobId}": {
  "delete": {
    "operationId": "deleteSchedule",
    "summary": "Delete the schedule with the specified ID.",
    "parameters": [
      {
        "name": "Authorization",
        "required": false,
        "in": "header",
        "schema": {
          "type": "string"
        }
      }
    ]
  }
}

```

```

    }
  },
  {
    "name": "catalogId",
    "required": true,
    "in": "path",
    "description": "The Catalog ID for which you want to delete a schedule.",
    "schema": {
      "type": "string"
    }
  },
  {
    "name": "modelId",
    "required": true,
    "in": "path",
    "description": "The Model ID for which you want to delete a schedule.",
    "schema": {
      "type": "string"
    }
  },
  {
    "name": "jobId",
    "required": true,
    "in": "path",
    "description": "The Job ID for which you want to delete a schedule.",
    "schema": {
      "type": "string"
    }
  }
],
"responses": {
  "200": {
    "description": "Success.",
    "content": {
      "text/plain": {
        "schema": {
          "type": "string",
          "example": "successfully deleted jobId: [aa5413f8-b87b-4b1a-a384-9a0408c85d2e]"
        }
      }
    }
  },
  "400": {
    "description": "Invalid request for example: Invalid parameters, missing parameters."
  },
  "401": {
    "description": "Unauthorized. This error will occur if calling user is not authenticated."
  },
  "403": {
    "description": "Forbidden. This error will occur if calling user does not have access right for the API."
  },
  "404": {
    "description": "Not found. The server cannot find the requested resource."
  },
  "500": {
    "description": "A server error occurred. Check the logs for more details."
  }
},
"tags": [
  "Aggregate Creation Schedule"
],
"security": [
  {
    "bearer": []
  }
]

```

```

    }
  ]
}
},
"/v1/public/aggregates-creation/builds/history": {
  "get": {
    "operationId": "getAllCreationSchedules",
    "summary": "Get all aggregate creation schedules.",
    "parameters": [
      {
        "name": "Authorization",
        "required": false,
        "in": "header",
        "schema": {
          "type": "string"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "Success.",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/AggregateBuildHistoryResponseDto"
            }
          }
        }
      },
      "400": {
        "description": "Invalid request for example: Invalid parameters, missing parameters."
      },
      "401": {
        "description": "Unauthorized. This error will occur if calling user is not authenticated."
      },
      "403": {
        "description": "Forbidden. This error will occur if calling user does not have access right for the API."
      },
      "404": {
        "description": "Not found. The server cannot find the requested resource."
      },
      "500": {
        "description": "A server error occurred. Check the logs for more details."
      }
    },
    "tags": [
      "Aggregate Creation Schedule"
    ],
    "security": [
      {
        "bearer": []
      }
    ]
  }
},
"/v1/public/catalogs/{catalogId}/models/{modelId}/aggregates-creation/execute": {
  "post": {
    "operationId": "triggerNewJob",
    "summary": "Trigger a new aggregate creation job.",
    "parameters": [
      {
        "name": "Authorization",
        "required": false,
        "in": "header",

```

```

    "schema": {
      "type": "string"
    }
  },
  {
    "name": "catalogId",
    "required": true,
    "in": "path",
    "description": "The Catalog ID for which you want to trigger a new job.",
    "schema": {
      "type": "string"
    }
  },
  {
    "name": "modelId",
    "required": true,
    "in": "path",
    "description": "The Model ID for which you want to trigger a new job.",
    "schema": {
      "type": "string"
    }
  }
],
"responses": {
  "201": {
    "description": "Success.",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/AggregateTriggerNewJobResponseDto"
        }
      }
    }
  },
  "400": {
    "description": "Invalid request for example: Invalid parameters, missing parameters."
  },
  "401": {
    "description": "Unauthorized. This error will occur if calling user is not authenticated."
  },
  "403": {
    "description": "Forbidden. This error will occur if calling user does not have access right for the API."
  },
  "404": {
    "description": "Not found. The server cannot find the requested resource."
  },
  "500": {
    "description": "A server error occurred. Check the logs for more details."
  }
},
"tags": [
  "Aggregate Creation Schedule"
],
"security": [
  {
    "bearer": []
  }
]
},
"/v1/public/aggregates": {
  "get": {
    "operationId": "aggregate",
    "summary": "Obtain details about aggregate definitions. You can query by ID, catalog, catalog id, model ID, status,

```

```

and subtype.",
  "parameters": [
    {
      "name": "Authorization",
      "required": false,
      "in": "header",
      "schema": {
        "type": "string"
      }
    },
    {
      "name": "id",
      "required": false,
      "in": "query",
      "schema": {
        "type": "string"
      }
    },
    {
      "name": "catalogId",
      "required": false,
      "in": "query",
      "schema": {
        "type": "string"
      }
    },
    {
      "name": "catalog",
      "required": false,
      "in": "query",
      "schema": {
        "type": "string"
      }
    },
    {
      "name": "modelId",
      "required": false,
      "in": "query",
      "schema": {
        "type": "string"
      }
    },
    {
      "name": "status",
      "required": false,
      "in": "query",
      "schema": {
        "enum": [
          "active",
          "invalid",
          "pending"
        ],
        "type": "string"
      }
    },
    {
      "name": "subtype",
      "required": false,
      "in": "query",
      "schema": {
        "enum": [
          "demand_defined",
          "prediction_defined",
          "manual",

```



```

    "hinted"
  ],
  "type": "string"
}
],
"responses": {
  "200": {
    "description": "Array with aggregate definitions",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/AggregateDefinitionsResponseDto"
        }
      }
    }
  },
  "400": {
    "description": "Invalid request for example: Invalid parameters, missing parameters."
  },
  "401": {
    "description": "Unauthorized. This error will occur if calling user is not authenticated."
  },
  "403": {
    "description": "Forbidden. This error will occur if calling user does not have access right for the API."
  },
  "404": {
    "description": "Not found. The server cannot find the requested resource."
  },
  "500": {
    "description": "A server error occurred. Check the logs for more details."
  }
},
"tags": [
  "Aggregate Details"
],
"security": [
  {
    "bearer": []
  }
]
},
"/v1/public/aggregates/instances": {
  "get": {
    "operationId": "aggregate-instances",
    "summary": "Obtain details about aggregate instances. You can query by definition ID, catalog ID, model ID, status, batch ID, table name, start time, and end time.",
    "parameters": [
      {
        "name": "Authorization",
        "required": false,
        "in": "header",
        "schema": {
          "type": "string"
        }
      },
      {
        "name": "definitionId",
        "required": false,
        "in": "query",
        "schema": {
          "type": "string"
        }
      }
    ]
  }
}

```

```

    },
    {
      "name": "catalogId",
      "required": false,
      "in": "query",
      "schema": {
        "type": "string"
      }
    },
    {
      "name": "modelId",
      "required": false,
      "in": "query",
      "schema": {
        "type": "string"
      }
    },
    {
      "name": "status",
      "required": false,
      "in": "query",
      "schema": {
        "enum": [
          "active",
          "invalid",
          "pending",
          "deleted"
        ],
        "type": "string"
      }
    },
    {
      "name": "batchId",
      "required": false,
      "in": "query",
      "schema": {
        "type": "string"
      }
    },
    {
      "name": "tableName",
      "required": false,
      "in": "query",
      "schema": {
        "type": "string"
      }
    },
    {
      "name": "startTime",
      "required": false,
      "in": "query",
      "schema": {
        "type": "string"
      }
    },
    {
      "name": "endTime",
      "required": false,
      "in": "query",
      "schema": {
        "type": "string"
      }
    }
  ],

```

```

"responses": {
  "200": {
    "description": "Array with aggregate instances",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/AggregateInstanceResponseDto"
        }
      }
    },
  },
  "400": {
    "description": "Invalid request for example: Invalid parameters, missing parameters."
  },
  "401": {
    "description": "Unauthorized. This error will occur if calling user is not authenticated."
  },
  "403": {
    "description": "Forbidden. This error will occur if calling user does not have access right for the API."
  },
  "404": {
    "description": "Not found. The server cannot find the requested resource."
  },
  "500": {
    "description": "A server error occurred. Check the logs for more details."
  }
},
"tags": [
  "Aggregate Details"
],
"security": [
  {
    "bearer": []
  }
]
},
"/v1/public/aggregates/definitions/{definitionId}/instances/{instanceId}": {
  "delete": {
    "operationId": "aggregate-delete",
    "summary": "Delete aggregate definition",
    "parameters": [
      {
        "name": "Authorization",
        "required": false,
        "in": "header",
        "schema": {
          "type": "string"
        }
      },
      {
        "name": "definitionId",
        "required": true,
        "in": "path",
        "schema": {
          "type": "string"
        }
      },
      {
        "name": "instanceId",
        "required": true,
        "in": "path",
        "schema": {
          "type": "string"
        }
      }
    ]
  }
}

```

```

    }
  },
  "responses": {
    "200": {
      "description": "Deactivate the specified aggregate instances.",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/DeactivateAggregateResponseDto"
          }
        }
      }
    },
    "400": {
      "description": "Invalid request for example: Invalid parameters, missing parameters."
    },
    "401": {
      "description": "Unauthorized. This error will occur if calling user is not authenticated."
    },
    "403": {
      "description": "Forbidden. This error will occur if calling user does not have access right for the API."
    },
    "404": {
      "description": "Not found. The server cannot find the requested resource."
    },
    "500": {
      "description": "A server error occurred. Check the logs for more details."
    }
  },
  "tags": [
    "Aggregate Details"
  ],
  "security": [
    {
      "bearer": []
    }
  ]
},
"/v1/public/aggregates/export/catalogs/{catalogId}/models/{modelId}": {
  "get": {
    "operationId": "export",
    "summary": "Read a specific Aggregate.",
    "description": "This API exports System-Defined aggregate definitions that may then be imported using the Imported to another system that has the same ModelID. Note that User-Defined Aggregates (UDA's) are not included in the export.",
    "parameters": [
      {
        "name": "catalogId",
        "required": true,
        "in": "path",
        "description": "The catalog ID (UUID) for which you are querying.",
        "schema": {
          "type": "string"
        }
      },
      {
        "name": "modelId",
        "required": true,
        "in": "path",
        "description": "The model ID (UUID) for which you are querying.",
        "schema": {
          "type": "string"
        }
      }
    ]
  }
}

```

```

    }
  },
  {
    "name": "Authorization",
    "required": false,
    "in": "header",
    "schema": {
      "type": "string"
    }
  }
],
"responses": {
  "200": {
    "description": "Success.",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/ExportAggregateResponseDto"
        }
      }
    }
  },
  "400": {
    "description": "Bad Request."
  },
  "401": {
    "description": "Unauthorized AtScale user."
  },
  "403": {
    "description": "User does not have permission to access the resource."
  },
  "500": {
    "description": "A server error occurred. Check the logs for more details."
  },
  "503": {
    "description": "This error will occur if Design Center APIs are not enabled."
  }
},
"tags": [
  "Aggregate"
],
"security": [
  {
    "bearer": []
  }
]
}
},
"/v1/public/aggregates/import/catalogs/{catalogId}/models/{modelId}": {
  "post": {
    "operationId": "import",
    "summary": "Uploads System-Defined Aggregate Definition.",
    "description": "This API imports System-Defined aggregate definitions exported from the UI or the /aggregates/export API. The Request Body is expected to match that of the /aggregates/export endpoint Response Body. The identical model, as defined by Model ID, must exist in the system for this call to succeed. This API does not support importing from newer versions to older versions of AtScale. For best results, Aggregate Definitions should be migrated between AtScale servers with identical version numbers.",
    "parameters": [
      {
        "name": "catalogId",
        "required": true,
        "in": "path",
        "description": "The catalog ID (UUID) for which you are querying.",
        "schema": {

```

```

    "type": "string"
  }
},
{
  "name": "modelId",
  "required": true,
  "in": "path",
  "description": "The model ID (UUID) for which you are querying.",
  "schema": {
    "type": "string"
  }
},
{
  "name": "Authorization",
  "required": false,
  "in": "header",
  "schema": {
    "type": "string"
  }
},
{
  "name": "importDistributionKey",
  "required": false,
  "in": "query",
  "description": "Whether to import distribution hints in aggregate logical plans (defaults to true if unspecified).",
  "schema": {
    "type": "boolean"
  }
},
{
  "name": "importPartitionKeys",
  "required": false,
  "in": "query",
  "description": "Whether to import partition hints in aggregate logical plans (defaults to true if unspecified).",
  "schema": {
    "type": "boolean"
  }
},
{
  "name": "importReplication",
  "required": false,
  "in": "query",
  "description": "Whether to import replication hints in aggregate logical plans (defaults to true if unspecified).",
  "schema": {
    "type": "boolean"
  }
},
{
  "name": "connectionRemap",
  "required": false,
  "in": "query",
  "description": "originalConnId:newConnId (no remap if unspecified). Pass multiple connection Remap variables to perform multiple remap operations.",
  "schema": {
    "type": "string"
  }
}
],
"requestBody": {
  "required": true,
  "content": {
    "application/json": {
      "schema": {
        "$ref": "#/components/schemas/ImportAggregateRequestDto"
      }
    }
  }
}

```

```

    }
  }
},
"responses": {
  "200": {
    "description": "Success.",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/ImportAggregateResponseDto"
        }
      }
    }
  },
  "400": {
    "description": "Bad Request."
  },
  "401": {
    "description": "Unauthorized AtScale user."
  },
  "403": {
    "description": "User does not have permission to access the resource."
  },
  "500": {
    "description": "A server error occurred. Check the logs for more details."
  },
  "503": {
    "description": "This error will occur if Design Center APIs are not enabled"
  }
},
"tags": [
  "Aggregate"
],
"security": [
  {
    "bearer": []
  }
]
},
"/v1/public/catalogs": {
  "get": {
    "operationId": "getCatalogs",
    "summary": "Get catalogs",
    "parameters": [
      {
        "name": "Authorization",
        "required": false,
        "in": "header",
        "schema": {
          "type": "string"
        }
      }
    ]
  },
  "responses": {
    "200": {
      "description": "Array of published catalogs",
      "content": {
        "application/json": {
          "schema": {
            "type": "array",
            "items": {
              "$ref": "#/components/schemas/GetCatalogsResponseDto"
            }
          }
        }
      }
    }
  }
}

```

```

    }
  }
}
},
"400": {
  "description": "Invalid request for example: Invalid parameters, missing parameters."
},
"401": {
  "description": "Unauthorized. This error will occur if calling user does not have access right for the API."
},
"500": {
  "description": "A server error occurred. Check the logs for more details."
}
},
"tags": [
  "Catalogs"
],
"security": [
  {
    "bearer": []
  }
]
}
},
"info": {
  "title": "AtScale API",
  "description": "NestJS API for AtScale",
  "version": "1.0",
  "contact": {}
},
"tags": [],
"servers": [],
"components": {
  "securitySchemes": {
    "bearer": {
      "scheme": "bearer",
      "bearerFormat": "JWT",
      "type": "http"
    }
  }
},
"schemas": {
  "AggregateCreationScheduleResponseDto": {
    "type": "object",
    "properties": {
      "jobId": {
        "type": "string",
        "format": "uuid"
      },
      "catalogId": {
        "type": "string",
        "format": "uuid"
      },
      "modelId": {
        "type": "string",
        "format": "uuid"
      },
      "fullBuild": {
        "type": "boolean"
      },
      "timeExpression": {
        "type": "string"
      }
    }
  },

```



```

"createdByUser": {
  "type": "string"
},
"createdAt": {
  "type": "string",
  "format": "date-time",
  "nullable": true
},
"latestExecution": {
  "type": "object",
  "properties": {
    "id": {
      "type": "string",
      "format": "uuid"
    },
    "jobId": {
      "type": "string",
      "format": "uuid"
    },
    "executedAt": {
      "type": "string",
      "format": "date-time",
      "nullable": true
    }
  },
  "required": [
    "id",
    "jobId",
    "executedAt"
  ]
},
"path": {
  "type": "string"
}
},
"required": [
  "jobId",
  "catalogId",
  "modelId",
  "fullBuild",
  "timeExpression",
  "createdByUser",
  "createdAt",
  "path"
]
},
"AggregateCreationScheduleRequestDto": {
  "type": "object",
  "properties": {
    "timeExpression": {
      "type": "string",
      "description": "When the schedule should be executed. Defined similarly to Cron job time format, with one more field (seconds) in the beginning. For example: 0 25 06 ? * *"
    }
  },
  "required": [
    "timeExpression"
  ]
},
"AggregateScheduleHistoryResponseDto": {
  "type": "object",
  "properties": {
    "history": {
      "type": "array",

```

```

    "items": {
      "type": "object",
      "properties": {
        "id": {
          "type": "string",
          "format": "uuid"
        },
        "jobId": {
          "type": "string",
          "format": "uuid"
        },
        "executedAt": {
          "type": "string",
          "format": "date-time",
          "nullable": true
        }
      },
      "required": [
        "id",
        "jobId",
        "executedAt"
      ]
    },
    "jobId": {
      "type": "string",
      "format": "uuid"
    },
    "required": [
      "history",
      "jobId"
    ]
  },
  "AggregateBuildHistoryResponseDto": {
    "type": "object",
    "properties": {
      "data": {
        "type": "array",
        "items": {
          "type": "object",
          "properties": {
            "id": {
              "type": "string",
              "format": "uuid"
            },
            "catalogId": {
              "type": "string",
              "format": "uuid"
            },
            "modelId": {
              "type": "string",
              "format": "uuid"
            },
            "createDate": {
              "type": "string",
              "format": "date-time",
              "nullable": true
            },
            "startTime": {
              "type": "string",
              "format": "date-time",
              "nullable": true
            },

```

```

    "endTime": {
      "type": "string",
      "format": "date-time",
      "nullable": true
    },
    "status": {
      "type": "string"
    },
    "estimateTime": {
      "type": "number"
    },
    "exceptionMessage": {
      "type": "string"
    },
    "exceptionClass": {
      "type": "string"
    },
    "batchType": {
      "type": "string"
    }
  },
  "required": [
    "id",
    "catalogId",
    "modelId",
    "createDate",
    "startTime",
    "endTime",
    "status",
    "estimateTime",
    "exceptionMessage",
    "exceptionClass",
    "batchType"
  ]
}
},
"required": [
  "data"
]
},
"AggregateTriggerNewJobResponseDto": {
  "type": "object",
  "properties": {
    "success": {
      "type": "boolean"
    }
  },
  "required": [
    "success"
  ]
},
"AggregateDefinitionsResponseDto": {
  "type": "object",
  "properties": {
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "id": {
            "type": "string",
            "minLength": 1,
            "format": "uuid"
          }
        }
      }
    }
  }
}

```

```

},
"name": {
  "type": "string",
  "minLength": 1
},
"type": {
  "type": "string",
  "enum": [
    "system_defined",
    "user_defined"
  ]
},
},
"subtype": {
  "type": "string",
  "minLength": 1
},
},
"incremental": {
  "type": "boolean"
},
},
"catalogId": {
  "type": "string",
  "minLength": 1
},
},
"catalogCaption": {
  "type": "string"
},
},
"modelId": {
  "type": "string",
  "minLength": 1
},
},
"connectionId": {
  "type": "string",
  "minLength": 1
},
},
"importedAt": {
  "type": "string"
},
},
"systemDefinedWithDistinctCount": {
  "type": "boolean"
},
},
"promoted": {
  "type": "boolean"
},
},
"latestInstance": {
  "type": "object",
  "properties": {
    "id": {
      "type": "string",
      "minLength": 1,
      "format": "uuid"
    },
    "status": {
      "type": "string",
      "enum": [
        "done",
        "invalid",
        "new",
        "inprogress",
        "unreliable",
        "cancelled",
        "error",
        "success",
        "deleted",
        "running",

```

```

    "active",
    "failed",
    "pending"
  ]
},
"message": {
  "type": "string"
},
"tableName": {
  "type": "string"
},
"tableSchema": {
  "type": "string"
},
"batchId": {
  "type": "string",
  "format": "uuid"
},
"stats": {
  "type": "object",
  "properties": {
    "materializationStartTime": {
      "type": "string",
      "minLength": 1
    },
    "materializationEndTime": {
      "type": "string"
    },
    "buildDuration": {
      "type": "number"
    },
    "numberOfRows": {
      "type": "number"
    }
  }
},
"required": [
  "materializationStartTime",
  "buildDuration"
]
},
"buildQueryId": {
  "type": "string",
  "format": "uuid"
},
"catalogId": {
  "type": "string",
  "minLength": 1,
  "format": "uuid"
},
"modelId": {
  "type": "string",
  "minLength": 1,
  "format": "uuid"
},
"definitionId": {
  "type": "string",
  "minLength": 1,
  "format": "uuid"
},
"connectionId": {
  "type": "string",
  "minLength": 1
}
},

```

```

"required": [
  "id",
  "status",
  "stats",
  "catalogId",
  "modelId",
  "definitionId",
  "connectionId"
]
},
"activeInstance": {
  "type": "object",
  "properties": {
    "id": {
      "type": "string",
      "minLength": 1,
      "format": "uuid"
    },
    "status": {
      "type": "string",
      "enum": [
        "done",
        "invalid",
        "new",
        "inprogress",
        "unreliable",
        "cancelled",
        "error",
        "success",
        "deleted",
        "running",
        "active",
        "failed",
        "pending"
      ]
    }
  }
},
"message": {
  "type": "string"
},
"tableName": {
  "type": "string"
},
"tableSchema": {
  "type": "string"
},
"batchId": {
  "type": "string",
  "format": "uuid"
},
"stats": {
  "type": "object",
  "properties": {
    "materializationStartTime": {
      "type": "string",
      "minLength": 1
    },
    "materializationEndTime": {
      "type": "string"
    },
    "buildDuration": {
      "type": "number"
    },
    "numberOfRows": {
      "type": "number"
    }
  }
}

```

```

    }
  },
  "required": [
    "materializationStartTime",
    "buildDuration"
  ]
},
"buildQueryId": {
  "type": "string",
  "format": "uuid"
},
"catalogId": {
  "type": "string",
  "minLength": 1,
  "format": "uuid"
},
"modelId": {
  "type": "string",
  "minLength": 1,
  "format": "uuid"
},
"definitionId": {
  "type": "string",
  "minLength": 1,
  "format": "uuid"
},
"connectionId": {
  "type": "string",
  "minLength": 1
}
},
"required": [
  "id",
  "status",
  "stats",
  "catalogId",
  "modelId",
  "definitionId",
  "connectionId"
]
},
"stats": {
  "type": "object",
  "properties": {
    "createdAt": {
      "type": "string",
      "minLength": 1
    },
    "averageBuildDuration": {
      "type": "number"
    },
    "queryUtilization": {
      "type": "number"
    },
    "mostRecentQuery": {
      "type": "string"
    }
  }
},
"required": [
  "createdAt",
  "queryUtilization"
]
},
"attributes": {

```

```

    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "name": {
          "type": "string",
          "minLength": 1
        },
        "type": {
          "type": "string",
          "minLength": 1
        },
        "partitioned": {
          "type": "boolean"
        }
      },
      "required": [
        "name",
        "type",
        "partitioned"
      ]
    }
  },
  "required": [
    "id",
    "name",
    "type",
    "subtype",
    "incremental",
    "catalogId",
    "catalogCaption",
    "modelId",
    "connectionId",
    "systemDefinedWithDistinctCount",
    "promoted",
    "latestInstance",
    "activeInstance",
    "stats",
    "attributes"
  ]
}
},
"required": [
  "data"
]
},
"AggregateInstanceResponseDto": {
  "type": "object",
  "properties": {
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "id": {
            "type": "string",
            "minLength": 1,
            "format": "uuid"
          },
          "status": {
            "type": "string",
            "enum": [

```



```

    "done",
    "invalid",
    "new",
    "inprogress",
    "unreliable",
    "cancelled",
    "error",
    "success",
    "deleted",
    "running",
    "active",
    "failed",
    "pending"
  ]
},
"message": {
  "type": "string"
},
"tableName": {
  "type": "string"
},
"tableSchema": {
  "type": "string"
},
"batchId": {
  "type": "string",
  "format": "uuid"
},
"stats": {
  "type": "object",
  "properties": {
    "materializationStartTime": {
      "type": "string",
      "minLength": 1
    },
    "materializationEndTime": {
      "type": "string"
    },
    "buildDuration": {
      "type": "number"
    },
    "numberOfRows": {
      "type": "number"
    }
  }
},
"required": [
  "materializationStartTime",
  "buildDuration"
]
},
"buildQueryId": {
  "type": "string",
  "format": "uuid"
},
"catalogId": {
  "type": "string",
  "minLength": 1,
  "format": "uuid"
},
"modelId": {
  "type": "string",
  "minLength": 1,
  "format": "uuid"
},

```

```

    "definitionId": {
      "type": "string",
      "minLength": 1,
      "format": "uuid"
    },
    "connectionId": {
      "type": "string",
      "minLength": 1
    }
  },
  "required": [
    "id",
    "status",
    "stats",
    "catalogId",
    "modelId",
    "definitionId",
    "connectionId"
  ]
}
},
"required": [
  "data"
]
},
"DeactivateAggregateResponseDto": {
  "type": "object",
  "properties": {
    "deleted": {
      "type": "boolean"
    }
  },
  "required": [
    "deleted"
  ]
},
"ExportAggregateResponseDto": {
  "type": "object",
  "properties": {
    "atScaleExportVersion": {
      "type": "string",
      "description": "The AtScale system version from which the export originated."
    },
    "exportSummary": {
      "type": "object",
      "properties": {
        "connectionIds": {
          "type": "object",
          "properties": {
            "count": {
              "type": "number"
            },
            "values": {
              "type": "array",
              "items": {
                "type": "string"
              }
            }
          }
        },
        "required": [
          "count"
        ]
      }
    }
  }
}

```

```

    },
    "required": [
      "connectionIds"
    ],
    "description": "Summarized information such as aggregate count, grouped by Connection ID."
  },
  "exportTimestamp": {
    "type": "string",
    "description": "Server-generated timestamp the export file was generated."
  },
  "exportCatalogId": {
    "type": "string",
    "description": "catalog ID"
  },
  "exportModelId": {
    "type": "string",
    "description": "Cube ID. Must exist on the target system."
  },
  "aggregates": {
    "type": "object",
    "properties": {
      "count": {
        "type": "number"
      },
      "values": {
        "type": "array",
        "items": {
          "type": "object",
          "properties": {
            "id": {
              "type": "string"
            },
            "userDefinedId": {
              "type": "string"
            },
            "userDefinedName": {
              "type": "string"
            },
            "triggeringQueryId": {
              "type": "string"
            },
            "environmentId": {
              "type": "string"
            },
            "catalogId": {
              "type": "string"
            },
            "planJson": {
              "type": "string"
            },
            "createdAt": {
              "nullable": true
            },
            "latestInstanceId": {
              "type": "string"
            },
            "baseType": {
              "nullable": true
            },
            "subType": {
              "nullable": true
            },
            "hintedDatasetId": {
              "type": "string"
            }
          }
        }
      }
    }
  }
}

```

```

    },
    "activeInstanceId": {
      "type": "string"
    },
    "connectionId": {
      "type": "string"
    },
    "promotedCopyOf": {
      "type": "boolean"
    },
    "promoted": {
      "type": "boolean"
    },
    "incidentalData": {
      "type": "object",
      "properties": {
        "type": {
          "type": "string"
        },
        "data-set-id": {
          "type": "string"
        }
      },
      "required": [
        "type",
        "data-set-id"
      ]
    },
  },
  "required": [
    "id",
    "userDefinedId",
    "userDefinedName",
    "triggeringQueryId",
    "environmentId",
    "catalogId",
    "planJson",
    "latestInstanceId",
    "hintedDatasetId",
    "activeInstanceId",
    "connectionId",
    "promotedCopyOf",
    "promoted",
    "incidentalData"
  ]
}
},
"required": [
  "count"
],
"description": "Aggregate Definitions. For internal use only."
}
},
"required": [
  "atScaleExportVersion",
  "exportSummary",
  "exportTimestamp",
  "exportCatalogId",
  "exportModelId",
  "aggregates"
]
},
"ImportAggregateRequestDto": {

```

```

"type": "object",
"properties": {
  "aggregates": {
    "type": "object",
    "properties": {
      "count": {
        "type": "number"
      },
      "values": {
        "type": "array",
        "items": {
          "type": "object",
          "properties": {
            "activeInstanceId": {
              "type": "string"
            },
            "baseType": {
              "type": "string"
            },
            "blocked": {
              "type": "boolean"
            },
            "connectionId": {
              "type": "string"
            },
            "createdAt": {
              "type": "string"
            },
            "modelId": {
              "type": "string"
            },
            "id": {
              "type": "string"
            },
            "latestInstanceId": {
              "type": "string"
            },
            "notes": {
              "type": "object",
              "properties": {
                "dimensional": {
                  "type": "boolean"
                }
              }
            },
            "required": [
              "dimensional"
            ]
          },
          "organizationId": {
            "type": "string"
          },
          "planJson": {
            "type": "object",
            "properties": {
              "model": {
                "type": "object",
                "properties": {
                  "id": {
                    "type": "string"
                  }
                }
              },
              "required": [
                "id"
              ]
            }
          }
        }
      }
    }
  }
}

```

```

    },
    "hints": {
      "type": "array",
      "items": {
        "nullable": true
      }
    },
    "selection": {
      "type": "object",
      "properties": {
        "columns": {
          "type": "array",
          "items": {
            "oneOf": [
              {
                "type": "object",
                "properties": {
                  "aggregation-type": {
                    "type": "object",
                    "properties": {
                      "type": {
                        "type": "string"
                      }
                    }
                  },
                },
                "required": [
                  "type"
                ]
              },
            ],
            "alias": {
              "type": "string"
            },
            "type": {
              "type": "string"
            },
            "value": {
              "type": "object",
              "properties": {
                "model": {
                  "type": "object",
                  "properties": {
                    "id": {
                      "type": "string"
                    }
                  }
                },
                "required": [
                  "id"
                ]
              }
            },
            "key": {
              "type": "object",
              "properties": {
                "key": {
                  "type": "object",
                  "properties": {
                    "model": {
                      "type": "string",
                      "format": "null",
                      "nullable": true
                    }
                  },
                  "id": {
                    "type": "string"
                  }
                },
                "type": {
                  "type": "string"
                }
              }
            }
          }
        }
      }
    }
  }

```

```

    }
  },
  "required": [
    "model",
    "id",
    "type"
  ]
},
"ref-path": {
  "type": "object",
  "properties": {
    "refs": {
      "type": "array",
      "items": {
        "nullable": true
      }
    },
    "type": {
      "type": "string"
    }
  },
  "required": [
    "refs",
    "type"
  ]
},
"required": [
  "key",
  "ref-path",
  "type"
]
},
"required": [
  "key",
  "ref-path",
  "type"
]
},
"required": [
  "aggregation-type",
  "alias",
  "type",
  "value"
]
},
{
  "type": "object",
  "properties": {
    "aggregation-type": {
      "type": "object",
      "properties": {
        "type": {
          "type": "string"
        }
      }
    }
  },
},

```

```

        "required": [
          "type"
        ]
      },
      "alias": {
        "type": "string"
      },
      "type": {
        "type": "string"
      },
      "value": {
        "type": "object",
        "properties": {
          "attribute": {
            "type": "object",
            "properties": {
              "model": {
                "type": "object",
                "properties": {
                  "id": {
                    "type": "string"
                  }
                }
              },
              "required": [
                "id"
              ]
            }
          },
          "name": {
            "type": "string"
          },
          "type": {
            "type": "string"
          }
        }
      },
      "required": [
        "model",
        "name",
        "type"
      ]
    },
    "type": {
      "type": "string"
    }
  },
  "required": [
    "attribute",
    "type"
  ]
}
},
"required": [
  "aggregation-type",
  "alias",
  "type",
  "value"
]
}
]
}
},
"from": {
  "type": "object",
  "properties": {
    "model": {

```



```

        "type": "object",
        "properties": {
          "id": {
            "type": "string"
          }
        },
        "required": [
          "id"
        ]
      },
      "type": {
        "type": "string"
      }
    },
    "required": [
      "model",
      "type"
    ]
  },
  "limit": {
    "type": "string",
    "format": "null",
    "nullable": true
  },
  "order": {
    "type": "array",
    "items": {
      "nullable": true
    }
  },
  "type": {
    "type": "string"
  }
},
"required": [
  "columns",
  "from",
  "limit",
  "order",
  "type"
]
},
"type": {
  "type": "string"
}
},
"required": [
  "model",
  "hints",
  "selection",
  "type"
]
},
"catalogId": {
  "type": "string"
},
"promoted": {
  "type": "boolean"
},
"subType": {
  "type": "string"
},
"triggeringQueryId": {
  "type": "string"
}

```

```

    }
  },
  "required": [
    "activeInstanceId",
    "baseType",
    "blocked",
    "connectionId",
    "createdAt",
    "modelId",
    "id",
    "latestInstanceId",
    "notes",
    "organizationId",
    "planJson",
    "catalogId",
    "promoted",
    "subType",
    "triggeringQueryId"
  ]
}
},
"required": [
  "count",
  "values"
]
},
"atScaleExportVersion": {
  "type": "string"
},
"exportModelId": {
  "type": "string"
},
"exportCatalogId": {
  "type": "string"
},
"exportSummary": {
  "type": "object",
  "properties": {
    "connectionIds": {
      "type": "object",
      "properties": {
        "count": {
          "type": "number"
        },
        "values": {
          "type": "array",
          "items": {
            "type": "string"
          }
        }
      }
    },
    "count": {
      "type": "number"
    },
    "values": {
      "type": "array",
      "items": {
        "type": "string"
      }
    }
  }
},
"required": [
  "count",
  "values"
]
},
"required": [
  "connectionIds"
]
},
"exportTimestamp": {
  "type": "string"
}

```

```

    }
  },
  "required": [
    "aggregates",
    "atScaleExportVersion",
    "exportModelId",
    "exportCatalogId",
    "exportSummary",
    "exportTimestamp"
  ]
},
"ImportAggregateResponseDto": {
  "type": "object",
  "properties": {
    "atScaleExportVersion": {
      "type": "string",
      "description": "The AtScale system version from which the export originated."
    },
    "exportCatalogId": {
      "type": "string",
      "description": "Catalog ID"
    },
    "importCatalogId": {
      "type": "string",
      "description": "ID of the Catalog to which the file was imported."
    },
    "modelId": {
      "type": "string",
      "description": "ID of the Model containing the aggregate definitions."
    },
    "numberOfDefinitionsImported": {
      "type": "number",
      "description": "Count of aggregate definitions that were imported."
    },
    "numberOfDefinitionsIgnored": {
      "type": "number",
      "description": "Count of aggregate definitions that were not imported."
    },
    "aggregates": {
      "type": "object",
      "properties": {
        "count": {
          "type": "number"
        },
        "values": {
          "type": "array",
          "items": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "newId": {
                "type": "string"
              },
              "imported": {
                "type": "boolean"
              },
              "reason": {
                "type": "string"
              }
            }
          }
        }
      }
    },
    "required": [
      "id",

```

```

        "imported"
      ]
    }
  },
  "required": [
    "count",
    "values"
  ],
  "description": "Count: count of the response Values array, Values[ id: Source Aggregate UUID, newID: New Aggregate UUID, Imported: True|False, reason: Human readable reason for Imported=False.]"
}
},
"required": [
  "atScaleExportVersion",
  "exportCatalogId",
  "importCatalogId",
  "modelId",
  "numberOfDefinitionsImported",
  "numberOfDefinitionsIgnored",
  "aggregates"
]
},
"GetCatalogsResponseDto": {
  "type": "object",
  "properties": {
    "id": {
      "type": "string",
      "minLength": 1,
      "format": "uuid"
    },
    "name": {
      "type": "string",
      "minLength": 1
    },
    "caption": {
      "type": "string",
      "minLength": 1
    },
    "publishedAt": {
      "type": "string",
      "format": "date-time",
      "nullable": true
    },
    "publishedBy": {
      "type": "string",
      "minLength": 1
    },
    "models": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "id": {
            "type": "string",
            "minLength": 1,
            "format": "uuid"
          },
          "name": {
            "type": "string",
            "minLength": 1
          },
          "caption": {
            "type": "string",

```

```
    "minLength": 1
  },
  "description": {
    "type": "string"
  },
  "connectionIds": {
    "type": "array",
    "items": {
      "type": "string"
    }
  },
  "required": [
    "id",
    "name",
    "caption",
    "connectionIds"
  ]
}
},
"required": [
  "id",
  "name",
  "caption",
  "publishedAt",
  "publishedBy",
  "models"
]
}
}
}
```