DataLens Dashboard Creation Guide

Document ID: PRD-DL-005 **Last Updated:** 2024-02-10 **Owner:** DataLens
Product Team **Classification:** Public

Overview

DataLens dashboards provide visual representations of your data. This guide
covers creating, configuring, and sharing dashboards.

Dashboard Basics

Creating a Dashboard

Via UI: 1. Navigate to Dashboards > New Dashboard 2. Enter dashboard
name and description 3. Select folder location 4. Click Create

Via API:

```
curl -X POST https://api.datalens.novatech.com/v1/dashboards \
  -H "Authorization: Bearer $TOKEN" \
  -d '{
    "name": "Sales Overview",
    "description": "Key sales metrics",
    "folder_id": "folder_abc"
  }'
```

Dashboard Structure

```
Dashboard
  Rows
      Panel 1 (Chart)
      Panel 2 (Table)
      Panel 3 (Stat)
  Variables (Filters)
  Time Range Picker
```

Adding Panels

Panel Types

Type	Use Case	Data Sources
Time Series	Trends over time	Metrics, Logs
Bar Chart	Comparisons	Any
Pie Chart	Proportions	Any
Table	Detailed data	Any
Stat	Single value	Any
Gauge	Progress/limits	Metrics
Heatmap	Density/patterns	Metrics
Geo Map	Location data	Any with coords
Text	Documentation	N/A

Creating a Time Series Panel

1. Click "Add Panel" > Visualization
2. Select data source (e.g., PostgreSQL)
3. Write query:

```sql
SELECT
  time_bucket('1 hour', created_at) AS time,
  count(*) AS orders
FROM orders
WHERE $__timeFilter(created_at)
GROUP BY 1
ORDER BY 1
```

4. Choose Time Series visualization
5. Configure axes and legend
6. Set title and save

Query Variables

Use variables in queries for dynamic filtering:

```sql
SELECT * FROM orders
WHERE region = '${region}'
  AND $__timeFilter(created_at)
```

Variable Types: | Type | Description | Example | |——|————-|————| | Query | Values from database | Region list | | Custom | Static list | ["US", "EU", "APAC"] | | Text | Free text input | Search term | | Interval | Time interval | 1h, 1d, 7d |

Panel Configuration

Display Options

Axes: - Y-axis: Scale (linear/log), min/max, units - X-axis: Time format, label rotation

Legend: - Position: Right, bottom, hidden - Values: Current, min, max, average

Colors: - Single color, palette, or thresholds - Color by value or series

Thresholds

Add visual indicators for important values:

```
{
  "thresholds": {
    "mode": "absolute",
    "steps": [
      {"value": 0, "color": "green"},
      {"value": 80, "color": "yellow"},
      {"value": 95, "color": "red"}
    ]
  }
}
```

Transformations

Apply transformations to query results:

Transformation	Description
Filter by name	Include/exclude series
Rename by regex	Modify series names
Reduce	Aggregate to single value
Join by field	Combine multiple queries

Transformation	Description
Add field	Calculate new fields
Group by	Aggregate data

Variables and Filters

Creating Variables

1. Dashboard Settings > Variables > New
2. Configure:

   - Name: `region`
   - Label: "Region"
   - Type: Query
   - Query: `SELECT DISTINCT region FROM customers`

3. Save

Variable Options

Option	Description
Multi-value	Allow multiple selections
Include All	Add "All" option
Default	Pre-selected value
Refresh	When to update values

Chained Variables

Create dependent variables:

```sql
-- First variable: region
SELECT DISTINCT region FROM customers

-- Second variable: city (depends on region)
SELECT DISTINCT city FROM customers
WHERE region = '${region}'
```

Time Range Controls

Time Picker Options

Option	Description
Relative	Last 5 minutes, 24 hours, 7 days
Absolute	Specific date range
Now delay	Account for data delay

Auto-Refresh

Set dashboard to refresh automatically: - Off, 5s, 10s, 30s, 1m, 5m, 15m, 30m, 1h

Time Zone

- Browser time (default)
- UTC
- Specific timezone

Layout and Organization

Grid Layout

- Dashboard uses 24-column grid
- Panels snap to grid
- Drag to resize and position

Rows

Group related panels:

```
Row: Sales Metrics
   Total Revenue (Stat)
   Orders (Time Series)
   Conversion Rate (Gauge)

Row: Regional Breakdown
   Revenue by Region (Bar)
   Top Cities (Table)
```

Collapsible Rows

Make rows collapsible for complex dashboards: 1. Edit row 2. Enable "Collapsible" 3. Set default state (open/closed)

Annotations

Creating Annotations

Mark important events on time series:

```sql
SELECT
  timestamp AS time,
  event_name AS text,
  severity AS tags
FROM events
WHERE $__timeFilter(timestamp)
```

Annotation Display

- Vertical lines on time series
- Hover for details
- Filter by tags

Alerting

Creating Alerts

1. Edit panel > Alert tab
2. Configure conditions:

   - Query: Which data to evaluate
   - Condition: When to trigger (e.g., avg() > 100)
   - Evaluate every: Check frequency
   - For: Duration before alerting

3. Configure notifications:

   - Email
   - Slack
   - PagerDuty
   - Webhook

Alert Example

```yaml
name: High Error Rate
conditions:
  - query: A
    reducer: avg
    evaluator: gt
    params: [5]   # Error rate > 5%
evaluate_every: 1m
for: 5m
notifications:
  - slack: #alerts
  - pagerduty: oncall-schedule
```

Sharing and Permissions

Dashboard Permissions

Level	Capabilities
Viewer	View dashboard
Editor	View + edit panels
Admin	Full control + permissions

Sharing Options

Within Organization: - Share link (requires login) - Add to team folder - Embed in other tools

External Sharing: - Public dashboard (no login) - Snapshot (point-in-time) - PDF export

Creating Snapshots

Capture dashboard at a moment: 1. Share > Snapshot 2. Set expiration (1 hour to never) 3. Share snapshot URL

Templates and Reuse

Dashboard Templates

Create reusable dashboard templates:

1. Create dashboard with variables
2. Export as JSON
3. Import and customize for different use cases

Panel Library

Save panels for reuse: 1. Edit panel > More > Save to library 2. Use in other dashboards 3. Updates sync across dashboards

---

Best Practices

Design Principles

1. **Clear Purpose:** Each dashboard should answer specific questions
2. **Progressive Detail:** Summary at top, details below
3. **Consistent Layout:** Similar data in similar positions
4. **Meaningful Titles:** Descriptive panel titles
5. **Documentation:** Add text panels for context

Performance

1. **Limit Time Range:** Default to reasonable range
2. **Optimize Queries:** Use indexes, aggregations
3. **Reduce Refresh Rate:** Only as fast as needed
4. **Limit Panel Count:** <20 panels per dashboard

Organization

```
Dashboards/
  Overview/
     Executive Summary
     System Health
  Teams/
     Engineering/
```

```
      Sales/
      Support/
   Services/
      API
      Database
      Queue
```

Troubleshooting

No Data Displayed

1. Check time range matches data
2. Verify query returns data
3. Check variable values
4. Review data source connection

Slow Dashboard

1. Analyze query performance
2. Reduce panel count
3. Increase refresh interval
4. Use query caching

Variables Not Working

1. Check variable syntax: `${variable}`
2. Verify variable has values
3. Check query uses variable correctly

Related Documents: Getting Started (PRD-DL-001), Query Language Reference (PRD-DL-010), Alerting Guide (PRD-DL-015)