DataLens Embedding Guide

Document ID: PRD-DL-030 **Last Updated:** 2024-02-28 **Owner:** DataLens Product Team **Classification:** Public

---

Overview

DataLens allows you to embed dashboards and panels in external applications, providing analytics capabilities wherever your users need them.

Embedding Options

Public Embedding

Share dashboards publicly without authentication.

Best for: - Public status pages - Marketing dashboards - Open data visualizations

Authenticated Embedding

Embed with user authentication and permissions.

Best for: - Customer-facing analytics - Internal portals - Partner dashboards

Signed Embedding

Server-side signed URLs for secure embedding.

Best for: - SaaS applications - White-labeled analytics - Multi-tenant environments

---

Public Embedding

Enable Public Access

1. Open your dashboard
2. Click **Share** → **Public Link**
3. Toggle **Enable public access**
4. Copy the embed URL

Embed Code

```
<iframe
  src="https://datalens.novatech.com/public/d/abc123"
  width="100%"
  height="600"
  frameborder="0"
></iframe>
```

Options

Parameter	Description	Default
`theme`	light, dark	light
`refresh`	Auto-refresh interval (seconds)	none
`from`	Start time	dashboard default
`to`	End time	dashboard default
`hideHeader`	Hide dashboard header	false
`hideControls`	Hide time picker	false

Example with options:

```
<iframe
  src="https://datalens.novatech.com/public/d/abc123?theme=dark&refresh=60&hideHeader=true"
  width="100%"
  height="600"
  frameborder="0"
></iframe>
```

Authenticated Embedding

Setup

1. Go to **Settings → Embedding**
2. Enable **Authenticated Embedding**
3. Configure allowed domains
4. Generate embed secret

Embed with Token

```
// Server-side: Generate embed URL with token
const jwt = require('jsonwebtoken');

function generateEmbedToken(user, dashboard) {
  const payload = {
    sub: user.id,
    email: user.email,
    dashboard: dashboard.id,
    permissions: ['view'],
    exp: Math.floor(Date.now() / 1000) + (60 * 60), // 1 hour
  };

  return jwt.sign(payload, EMBED_SECRET);
}

const embedUrl = `https://datalens.novatech.com/embed/d/${dashboardId}?token=${token}`;
```

Client-Side Embedding

```
<iframe
  id="datalens-embed"
  src=""
  width="100%"
  height="600"
  frameborder="0"
></iframe>

<script>
  // Fetch embed URL from your server
  fetch('/api/embed-url?dashboard=abc123')
    .then(res => res.json())
    .then(data => {
      document.getElementById('datalens-embed').src = data.url;
```

```
    });
</script>
```

Signed Embedding

How It Works

1. User requests embedded content
2. Your server generates signed URL
3. DataLens validates signature
4. Content rendered with user context

Server-Side Implementation

```
const crypto = require('crypto');

function generateSignedEmbedUrl(options) {
  const {
    dashboardId,
    userId,
    userEmail,
    expiration,
    variables,
  } = options;

  const params = {
    dashboard: dashboardId,
    user_id: userId,
    user_email: userEmail,
    exp: expiration,
    vars: JSON.stringify(variables),
  };

  const queryString = new URLSearchParams(params).toString();
  const signature = crypto
    .createHmac('sha256', EMBED_SECRET)
    .update(queryString)
    .digest('hex');

  return `https://datalens.novatech.com/embed/d/${dashboardId}?${queryString}&sig=${signatu
}
```

```
// Usage
const embedUrl = generateSignedEmbedUrl({
  dashboardId: 'abc123',
  userId: 'user_456',
  userEmail: 'user@example.com',
  expiration: Date.now() + 3600000, // 1 hour
  variables: {
    customer_id: '789',
    region: 'us-west',
  },
});
```

Python Implementation

```
import hmac
import hashlib
import json
import time
from urllib.parse import urlencode

def generate_signed_embed_url(dashboard_id, user_id, user_email, variables=None):
    expiration = int(time.time()) + 3600  # 1 hour

    params = {
        'dashboard': dashboard_id,
        'user_id': user_id,
        'user_email': user_email,
        'exp': expiration,
        'vars': json.dumps(variables or {}),
    }

    query_string = urlencode(params)
    signature = hmac.new(
        EMBED_SECRET.encode(),
        query_string.encode(),
        hashlib.sha256
    ).hexdigest()

    return f"https://datalens.novatech.com/embed/d/{dashboard_id}?{query_string}&sig={signat
```

Variable Injection

Dashboard Variables

Pass variables to filter dashboard data:

```
const embedUrl = generateSignedEmbedUrl({
  dashboardId: 'abc123',
  variables: {
    customer_id: '789',       // Filter by customer
    date_range: 'last_30d',   // Set time range
    region: 'us-west',        // Filter by region
  },
});
```

Using Variables in Queries

In your dashboard queries:

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

Variable Security

- Variables are validated against allowed values
- Prevent SQL injection with parameterized queries
- Restrict variable values in dashboard settings

Customization

Theming

```
<iframe
  src="https://datalens.novatech.com/embed/d/abc123?theme=custom"
></iframe>

<style>
  /* Custom CSS for embedded dashboard */
```

```
  :root {
    --datalens-bg: #1a1a2e;
    --datalens-text: #eee;
    --datalens-primary: #00d9ff;
  }
</style>
```

Custom Themes

Create custom themes in DataLens:

1. Go to **Settings → Themes**
2. Click **Create Theme**
3. Configure colors and fonts
4. Save and use theme ID in embed

White Labeling

Enterprise plans support white labeling: - Custom logo - Custom colors - Remove DataLens branding - Custom domain (CNAME)

JavaScript SDK

Installation

```
npm install @novatech/datalens-embed
```

Usage

```
import { DataLensEmbed } from '@novatech/datalens-embed';

const embed = new DataLensEmbed({
  container: '#dashboard-container',
  dashboardId: 'abc123',
  token: embedToken,
  theme: 'dark',
  variables: {
    customer_id: '789',
  },
});
```

```javascript
// Listen for events
embed.on('loaded', () => {
  console.log('Dashboard loaded');
});

embed.on('error', (error) => {
  console.error('Embed error:', error);
});

embed.on('variableChange', (variable, value) => {
  console.log(`Variable ${variable} changed to ${value}`);
});

// Update variables dynamically
embed.setVariable('customer_id', '456');

// Refresh data
embed.refresh();
```

React Component

```jsx
import { DataLensDashboard } from '@novatech/datalens-embed/react';

function Analytics({ customerId }) {
  return (
    <DataLensDashboard
      dashboardId="abc123"
      token={embedToken}
      variables={{ customer_id: customerId }}
      theme="dark"
      onLoaded={() => console.log('Loaded')}
      onError={(e) => console.error(e)}
    />
  );
}
```

Panel Embedding

Embed Single Panel

Embed individual panels instead of full dashboards:

```html
<iframe
  src="https://datalens.novatech.com/embed/p/panel123?dashboard=abc123"
  width="400"
  height="300"
  frameborder="0"
></iframe>
```

Multiple Panels

```html
<div class="analytics-grid">
  <iframe src=".../embed/p/revenue-chart?..."></iframe>
  <iframe src=".../embed/p/user-growth?..."></iframe>
  <iframe src=".../embed/p/conversion-rate?..."></iframe>
</div>
```

Security

Domain Restrictions

Configure allowed domains:

1. Go to **Settings → Embedding → Security**
2. Add allowed domains
3. Wildcards supported (*.example.com)

Token Security

- Keep embed secret secure
- Use short token expiration
- Rotate secrets periodically
- Log embed access

Row-Level Security

Filter data based on user:

```
SELECT *
FROM orders
WHERE tenant_id = '{{__user_tenant}}'
```

The `__user_tenant` is extracted from the signed token.

Content Security Policy

If using CSP, add DataLens domain:

```
<meta http-equiv="Content-Security-Policy"
  content="frame-src https://datalens.novatech.com">
```

Performance

Optimization Tips

1. **Use panel embedding** for faster loads
2. **Set appropriate refresh intervals**
3. **Limit data returned** with filters
4. **Use caching** for static data
5. **Lazy load** dashboards below the fold

Caching

Configure cache TTL:

```
const embedUrl = generateSignedEmbedUrl({
  dashboardId: 'abc123',
  cacheMinutes: 5, // Cache data for 5 minutes
});
```

Troubleshooting

Embed Not Loading

- Check domain is in allowed list
- Verify token hasn't expired
- Check browser console for errors
- Verify iframe permissions

Data Not Showing

- Verify variables are passed correctly
- Check user has data access
- Review query for errors
- Check data source connectivity

Styling Issues

- Use `height: 100%` on container
- Check for CSS conflicts
- Use isolated iframe sandbox

API Reference

Embed URL Parameters

Parameter	Description
`token`	Authentication token
`theme`	Theme name or ID
`from`	Start time (ISO 8601)
`to`	End time (ISO 8601)
`refresh`	Auto-refresh seconds
`hideHeader`	Hide header (true/false)
`hideControls`	Hide controls (true/false)
`vars`	JSON-encoded variables

SDK Methods

Method	Description
`setVariable(name, value)`	Update variable
`refresh()`	Refresh data
`destroy()`	Clean up embed
`on(event, callback)`	Event listener

Related Documents: Getting Started (PRD-DL-001), Dashboard Creation (PRD-DL-005), API Reference (PRD-DL-015)