



**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**

## Atelier 4 : Conception d'un Dashboard Observabilité avec Prometheus & Grafana

### Objectifs

À la fin de cet atelier, le participant sera capable de :

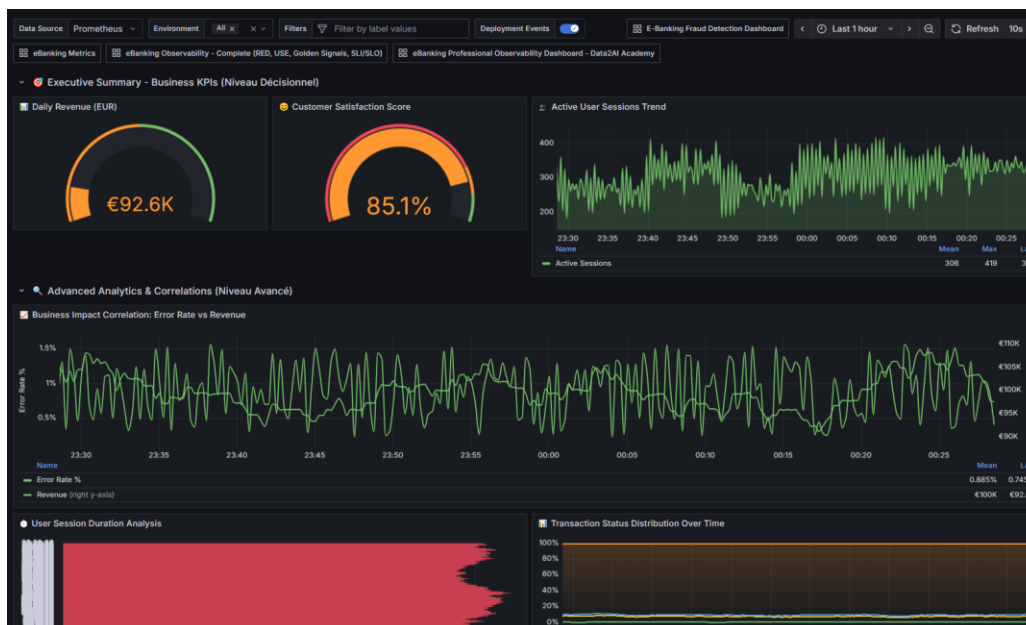
1. Configurer une datasource Prometheus et les variables (environment, datasource) nécessaires.
2. Comprendre et adapter les requêtes PromQL présentes dans le Dashboard fourni.
3. Personnaliser les panels (Gauges, TimeSeries, BarChart, Table, Text) selon le besoin métier.
4. Automatiser la collecte de métriques (rafraîchissement, annotations, alertes basiques) et sauvegarder le Dashboard.

---

### Aperçu du Dashboard



**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**



## Pré-requis

- Grafana installé et accessible.
- Prometheus avec métriques listées (simulateur). (Voir `simulate.sh` fourni).
- Accès admin dans Grafana pour créer dashboards et datasources.

## 🔧 1. Préparation & Environnement

### Étape 1.0 : Simuler les trafics de paiements

Place-toi dans ton repo d'atelier (ex : `~/observability-stack`) et lance le simulateur :

```
cd ~/observability-stack/payment-api-mock && chmod +x simulate.sh
```

```
sudo ./simulate.sh
```



**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**

Le simulateur doit pousser les métriques Prometheus suivantes (exemples trouvés dans le JSON) :

- ebanking\_daily\_revenue\_eur
- ebanking\_customer\_satisfaction\_score
- ebanking\_active\_sessions
- ebanking\_api\_errors\_total, ebanking\_api\_requests\_total
- ebanking\_session\_duration\_seconds\_bucket
- ebanking\_transactions\_processed\_total
- ebanking\_request\_duration\_seconds\_bucket

```
ubuntu@bhf-stack:~/observability-stack/payment-api-mock$ cd ~/observability-stack/payment-api-mock && chmod +x simulate.sh
sudo ./simulate.sh

=====
Payment API Simulation
=====
API URL:      http://localhost:8080/api/payments
Requests:    100
Mode:        normal
Max Concurrent: 5
Success Rate: 84%
Failure Rate: 1%
Pending Rate: 15%
=====

Checking API health...
API is healthy

Running NORMAL mode (standard traffic)

[1] ✓ Payment pay_1761608264443_2284 | 77.76 EUR | Customer: cust_00041 | .149039574s
[2] ✓ Payment pay_1761608267002_5227 | 47.39 EUR | Customer: cust_02587 | .228133814s
[3] ✓ Payment pay_1761608269813_7847 | 298.19 EUR | Customer: cust_00025 | .071175734s
[4] ✓ Payment pay_1761608271680_4557 | 61.15 EUR | Customer: cust_03102 | .238026274s
[5] ✗ HTTP 500 | 0.0 EUR | Customer: cust_00762 | Error: Unknown error | .488982172s
[6] ✓ Payment pay_1761608275534_6817 | 94.52 CHF | Customer: cust_01347 | .367736316s
[7] ✓ Payment pay_1761608278340_2871 | 42.34 GBP | Customer: cust_00025 | .155170195s
[8] ✓ Payment pay_1761608279700_3194 | 101.96 EUR | Customer: cust_04972 | .319927335s
```

## Étape 1.2 : Créer le Dashboard

1. Cliquez sur + → **Dashboard**
2. Un panel vide apparaît
3. Cliquez sur (**Paramètres**) pour définir :

Section	Champ	Valeur / Action
General	Title	Payment Analytics – Production Dashboard
	Description	Supervision e-Banking : volumes, latences, taux de succès, anomalies, satisfaction client.



**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**

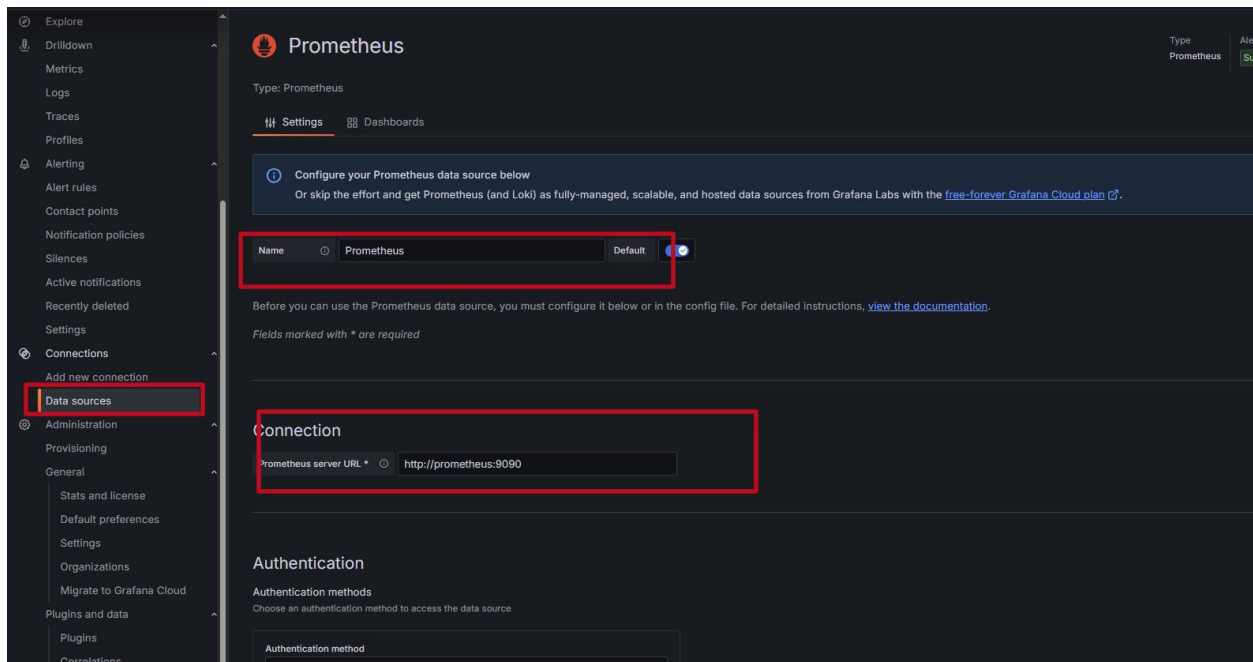
Section	Champ	Valeur / Action
	Tags	payments, observability, production, sla
	Editable	<input checked="" type="checkbox"/> Oui
Time options	Time zone	Browser Time
	Auto refresh	30s, 1m, 5m, 10m
Panel options	Graph tooltip	Default
	Preload panels	<input checked="" type="checkbox"/> Activé

## Étape 3: Vérifier la datasource Prometheus et variables

- Ouvrir Grafana → *Configuration* (gear icon) → *Data sources* → *Add data source* → choisir *Prometheus*.
  - URL: `http://<prometheus-host>:9090`
  - Access: *Server* (default)
  - Click **Save & Test**.



**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**



## 2. Créer la variable DS\_PROMETHEUS

- Grafana → *Dashboard* (nouveau) → *Settings* (gear) → *Variables* → *Add variable*.
- **Type: Datasource**
- **Name: DS\_PROMETHEUS**
- **Data source type: Prometheus**
- **Save.**

## 3. Créer la variable environment

- **Type: Query**
- **Name: environment**
- **Data source: \${DS\_PROMETHEUS}**
- **Query: label\_values(ebanking\_transactions\_processed\_total, environment)**
- **Multi: true, Include All: true, All value: .\***
- **Refresh: On Time Range Change.**

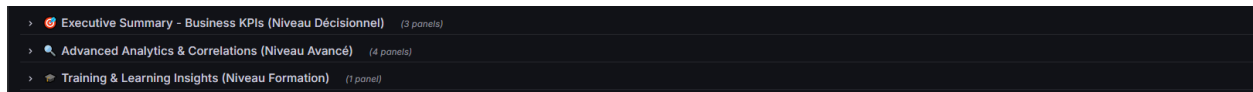
## 4. Adhoc filter (optionnel) : ajouter variable adhoc type **Ad hoc filters**.



**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**

## Étape 4: Créer la mise en page globale

1. Grafana -> *Create* -> *Dashboard* -> *Add an empty row* (tu vas créer 3-4 rows correspondant aux sections du JSON) :
  - Row 1: Executive Summary - Business KPIs (Niveau Décisionnel)
  - Row 2: Advanced Analytics & Correlations (Niveau Avancé)
  - Row 3: API Endpoint Performance Summary (SLO Dashboard)
  - Row 4: Training & Learning Insights (Niveau Formation)
2. Configurer *Time range* par défaut: now-1h → now (Dashboard settings -> Time range).
3. Refresh: 10s dans Dashboard settings -> General -> Refresh.



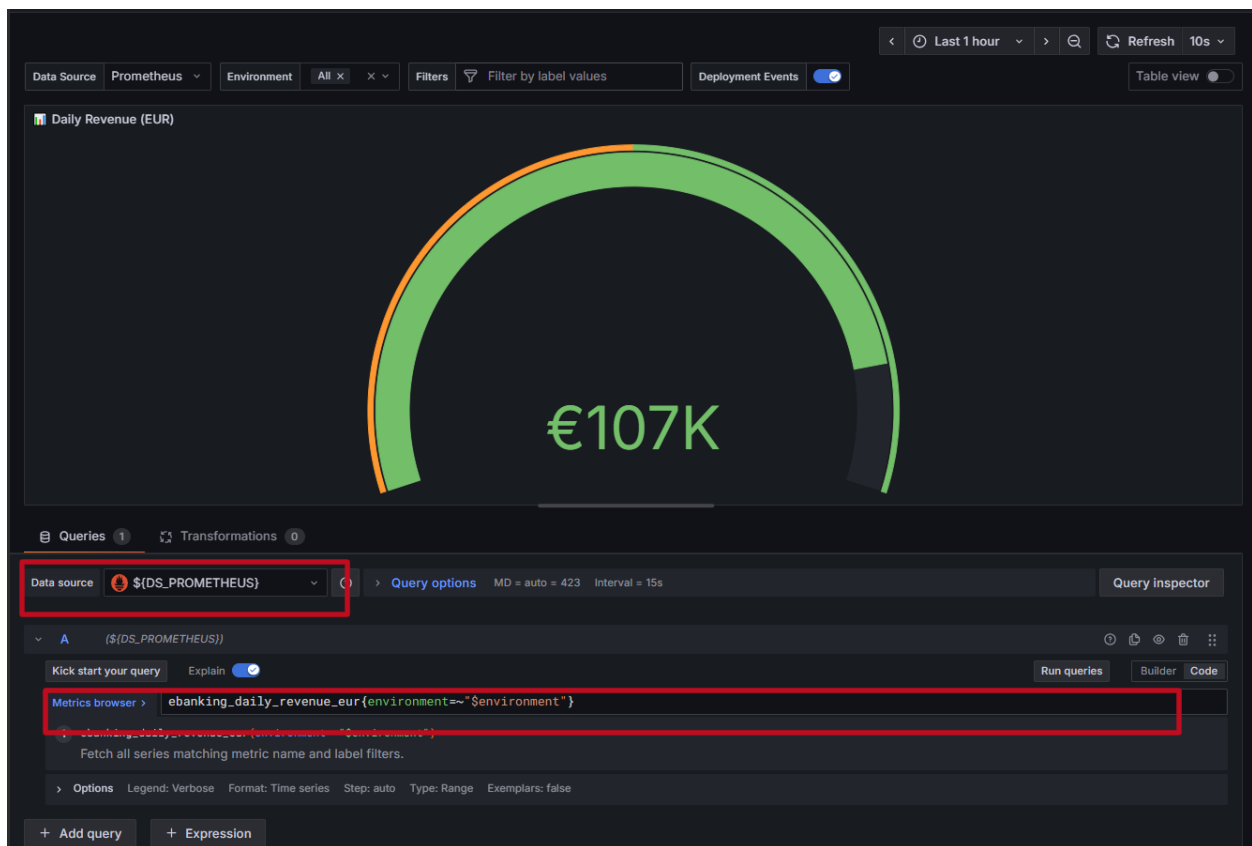
## Étape 5: Row 1 — Executive Summary: créer les 3 panels


### Panel A — Gauge: " Daily Revenue (EUR)"

1. Click *Add panel* → *Add new panel*.
2. Datasource: `${DS_PROMETHEUS}`.
3. Query (PromQL): `ebanking_daily_revenue_eur{environment=~"$environment"}` (met la variable `$environment` dans l'éditeur).



**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**

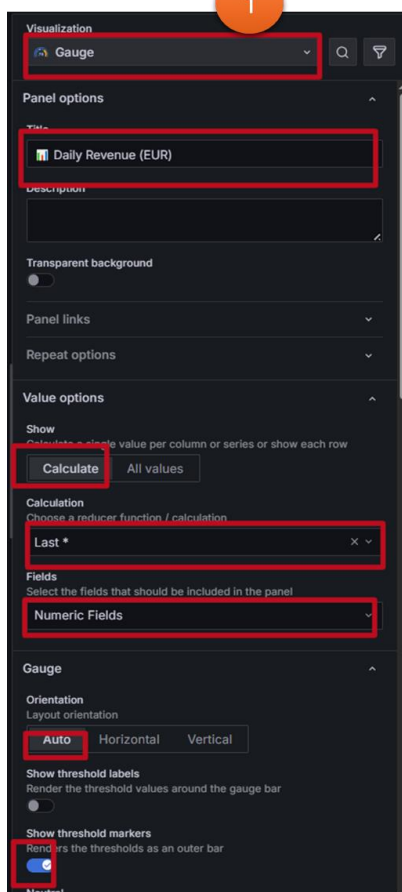


4. Visualisation: **Gauge**.
- 5.
6. Field → Unit: currencyEUR.
7. Field → Min/Max: (optional) laisser automatiques.
8. Field → Thresholds → Mode: absolute → Steps:
  - 0 → color red
  - 50000 → color orange
  - 100000 → color green
9. Options → Orientation: auto, Show threshold markers true.
10. Panel title:  Daily Revenue (EUR).
11. Save panel.

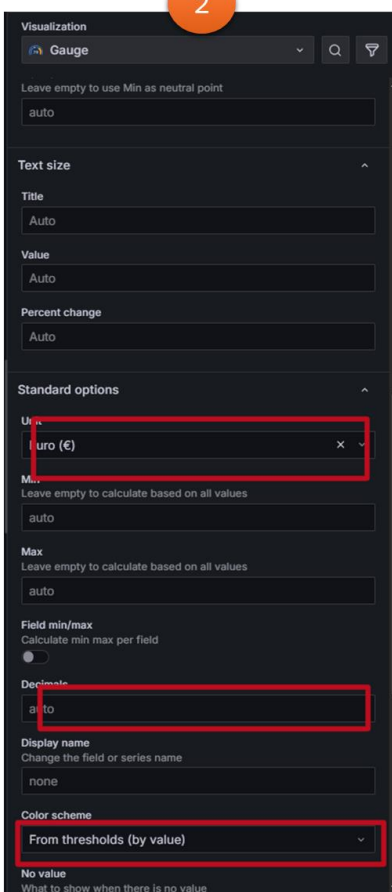


**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**

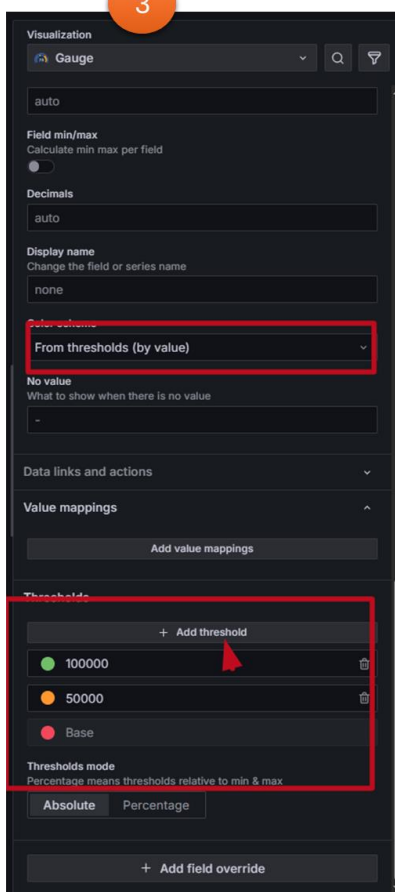
**1**



**2**



**3**



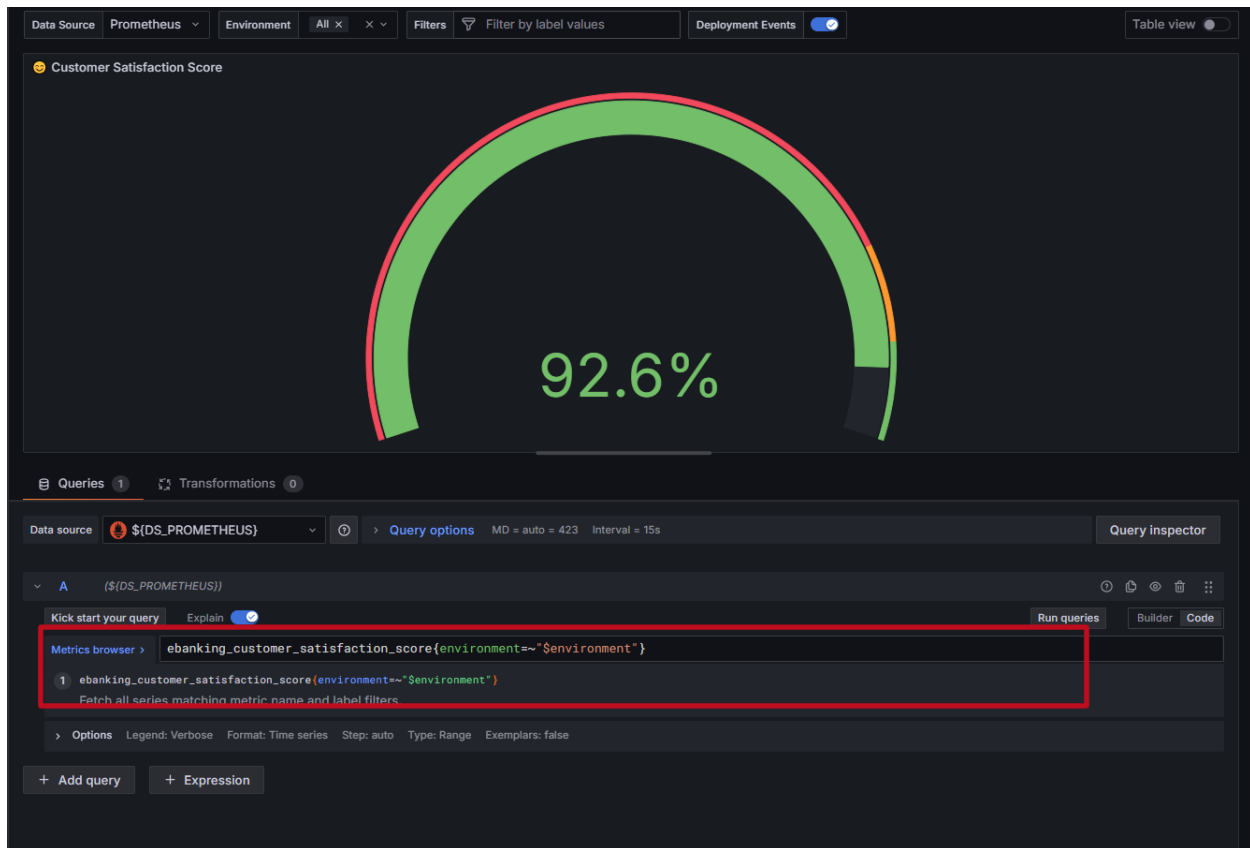
Conseil UI: Dans **Overrides** ajouter un override `byName: Revenue` si tu veux forcer axis placement ou unité alternative.

**Panel B — Gauge: "😊 Customer Satisfaction Score"**





**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**

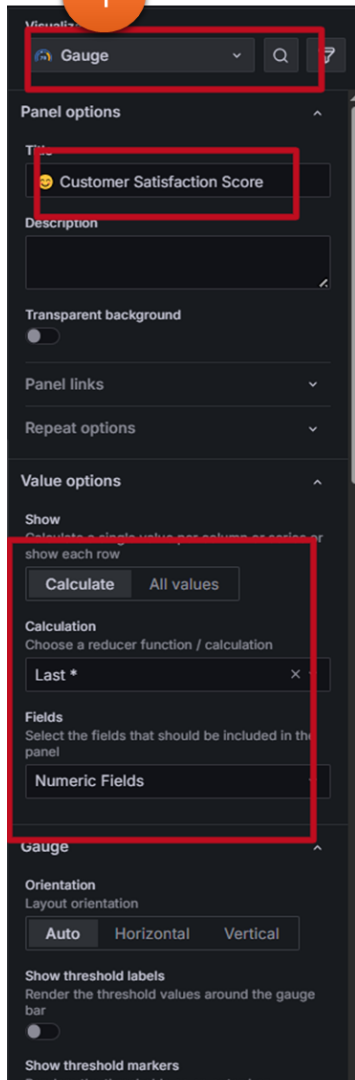


1. Add panel → Datasource: `${DS_PROMETHEUS}`.
2. Query: `ebanking_customer_satisfaction_score{environment=~"$environment"}`.
3. Visualisation: **Gauge**.
4. Field → Unit: percent, Min=0, Max=100.
5. Thresholds steps:
  - 0 → red
  - 80 → orange
  - 90 → green
6. Title: 😊 Customer Satisfaction Score.



**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**

**1**



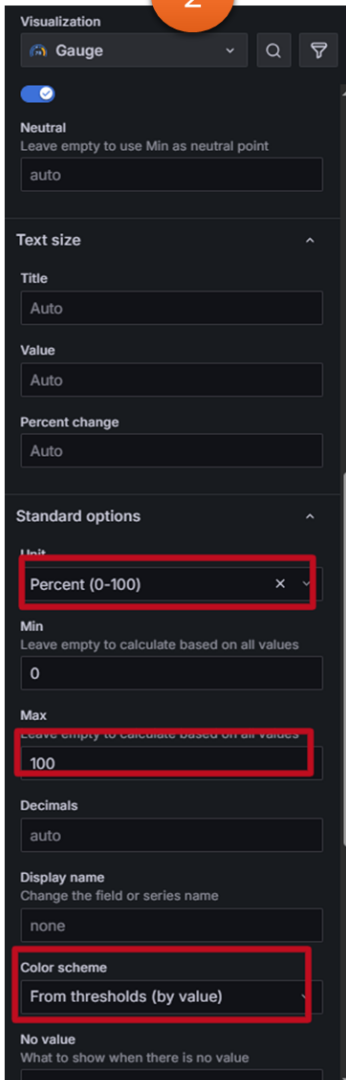
Panel options

Customer Satisfaction Score

Calculation: Last \*

Fields: Numeric Fields

**2**



Visualization

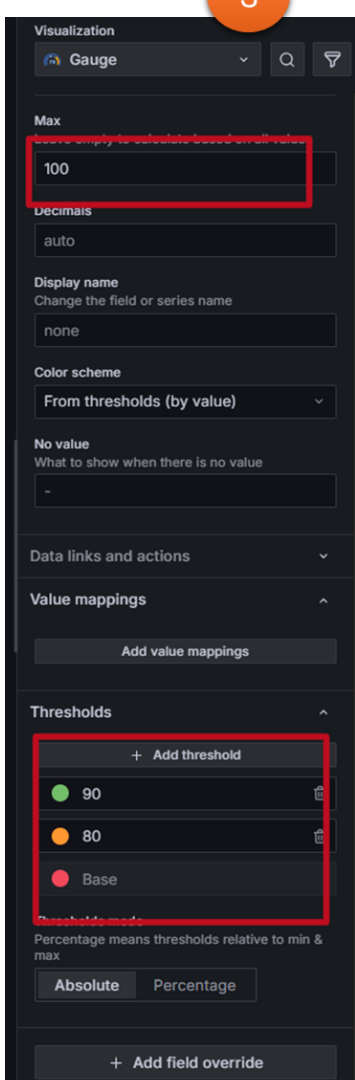
Unit: Percent (0-100)

Min: 0

Max: 100

Color scheme: From thresholds (by value)

**3**



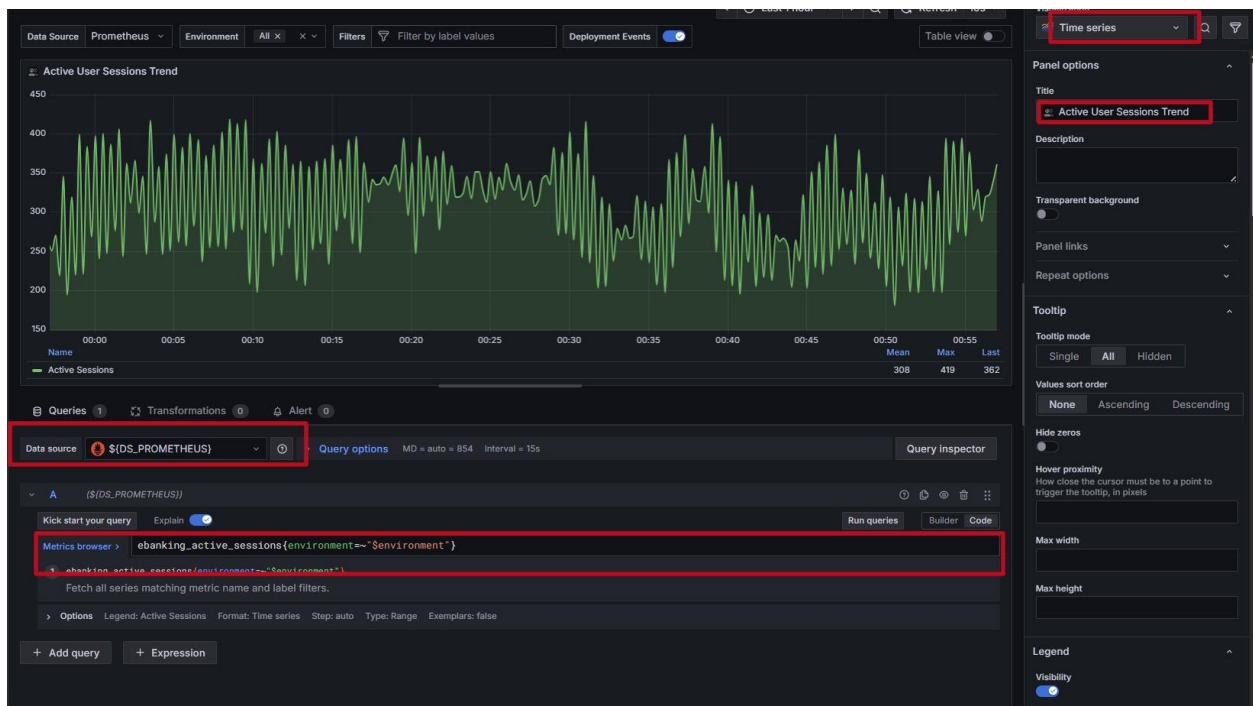
Thresholds


- 90
- 80
- Base

Panel C — Time Series: "👤 Active User Sessions Trend"



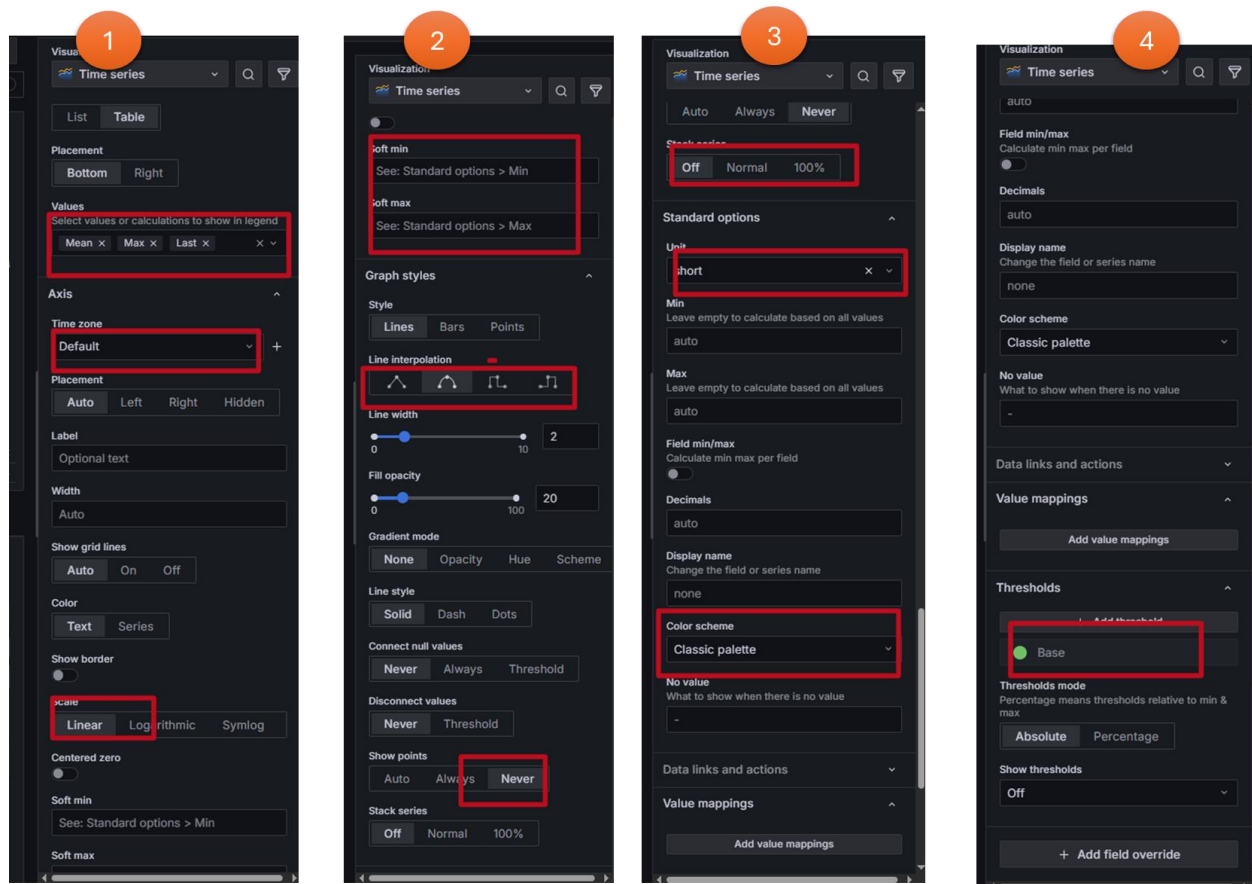
**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**



1. Add panel → Datasource `${DS_PROMETHEUS}`.
2. Query: `ebanking_active_sessions{environment=~\"$environment\"}`
3. Visualisation: **Time series**.
4. Options → Legend: set Mean, Max, Last in calcs, displayMode: table, placement: bottom.
5. Field → Unit: short.
6. Smooth lines: in Field custom → lineInterpolation: smooth and lineWidth: 2.
7. Title:  Active User Sessions Trend.



**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**



## 5) Row 2 — Advanced Analytics & Correlations

Panel: Time Series —  **Business Impact Correlation: Error Rate vs Revenue**



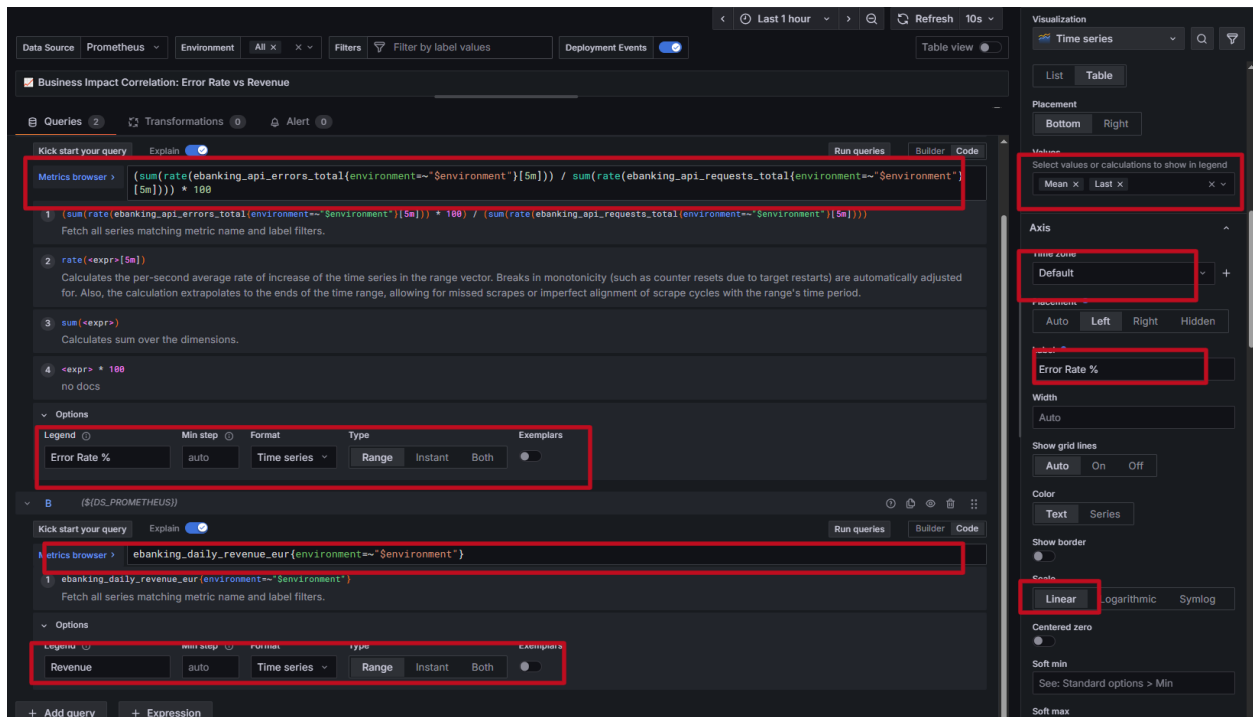
**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**



1. Add new panel.
2. Datasource: `${DS_PROMETHEUS}`.
3. Target A (Error Rate %):
  - Expr:
 
$$\frac{(\text{sum}(\text{rate}(\text{ebanking\_api\_errors\_total}\{\text{environment}=\sim\text{"\$environment"}\}[5\text{m}]))}{\text{sum}(\text{rate}(\text{ebanking\_api\_requests\_total}\{\text{environment}=\sim\text{"\$environment"}\}[5\text{m}]))} * 100$$
  - Legend: Error Rate %
4. Target B (Revenue):
  - Expr: `ebanking_daily_revenue_eur{environment=\sim\text{"\$environment"}}`
  - Legend: Revenue



**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**



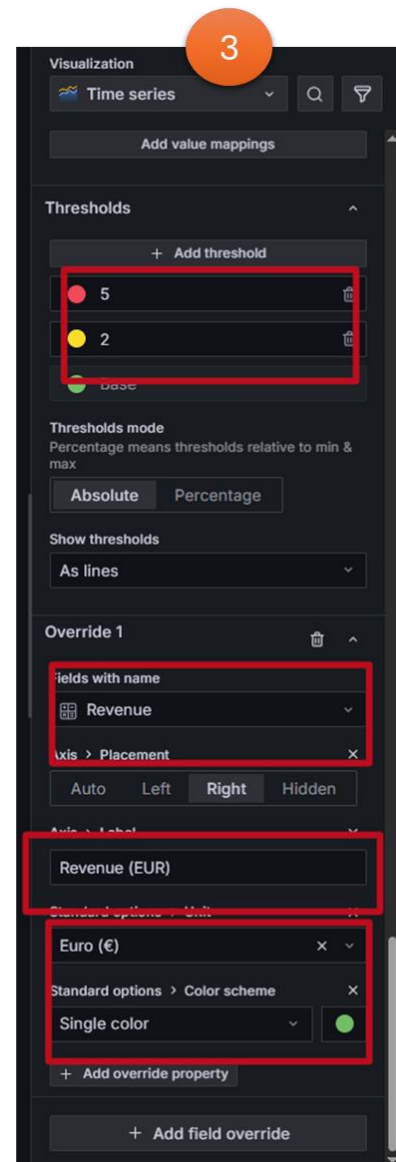
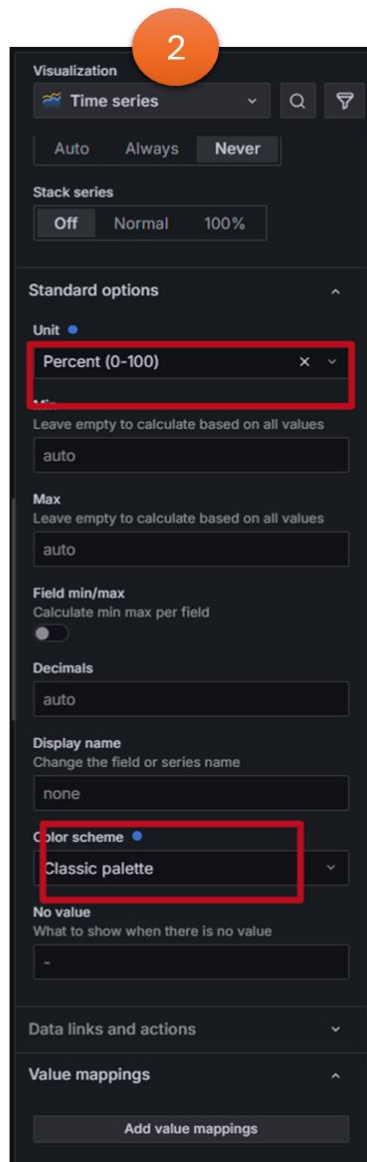
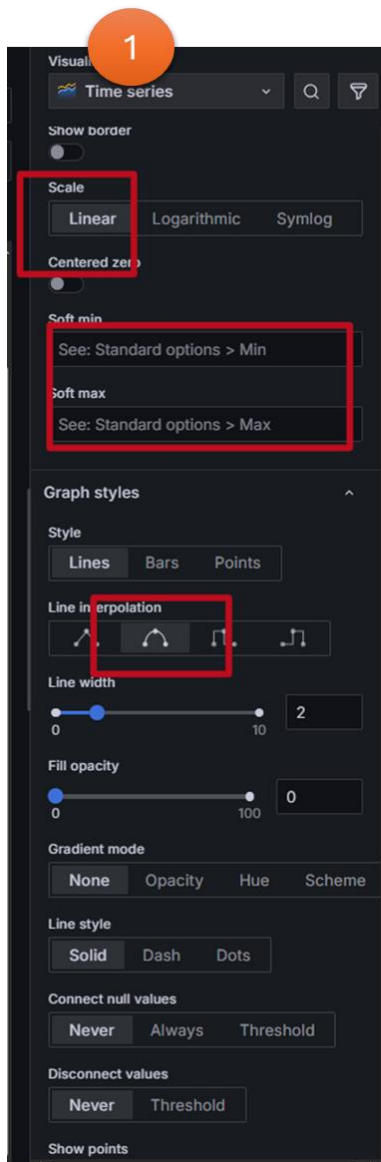
## 5. Visualization: Time series with dual axis

- Field overrides: match Revenue → set unit currencyEUR, axis placement right.
- Error Rate % unit: percent, axis left.






**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**



6. Legend: show mean and last.

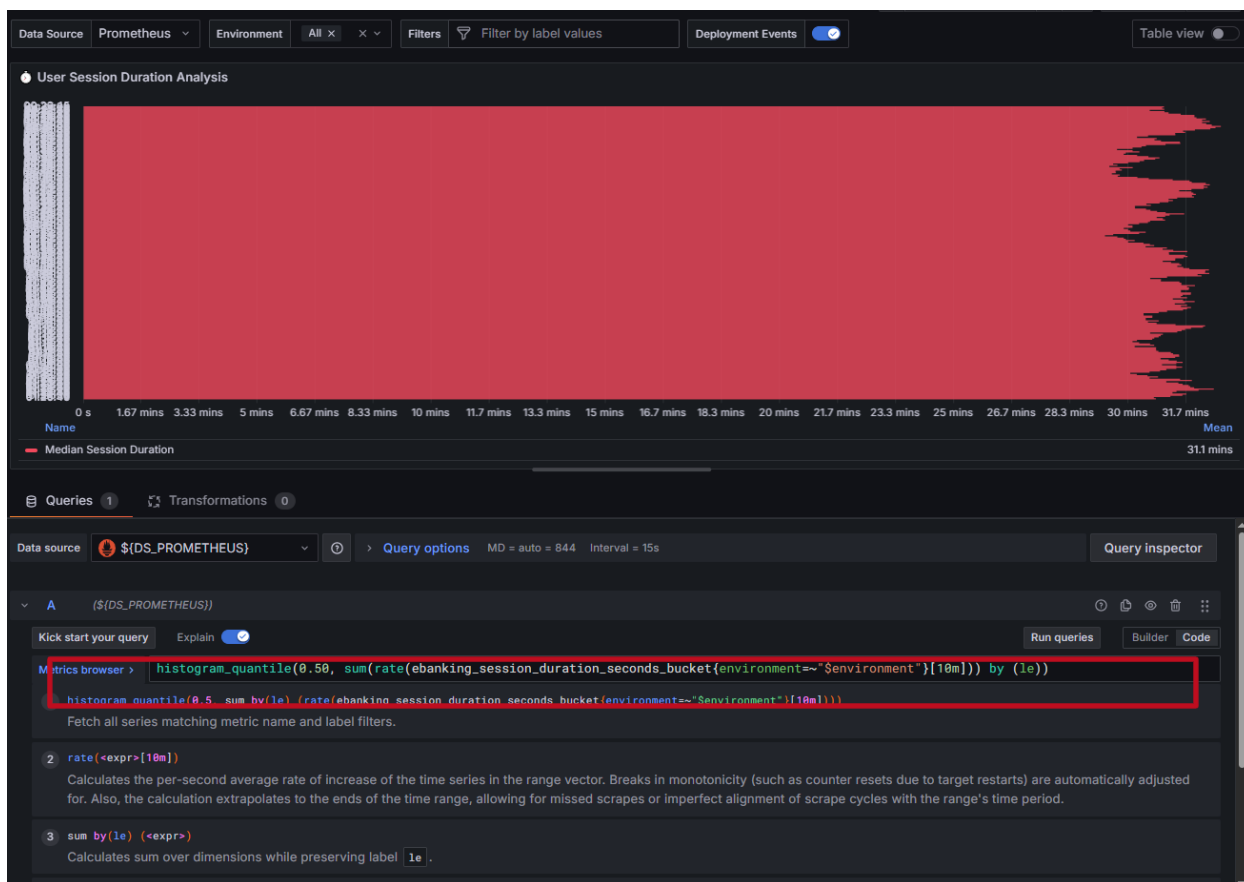
7. Title:  Business Impact Correlation: Error Rate vs Revenue.



**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**

UI tip: In the right pane → Field overrides → add matcher `byName` with the exact legend label to set axis/unit per series.

## 6) Row 3 — Performance & Distribution Panels



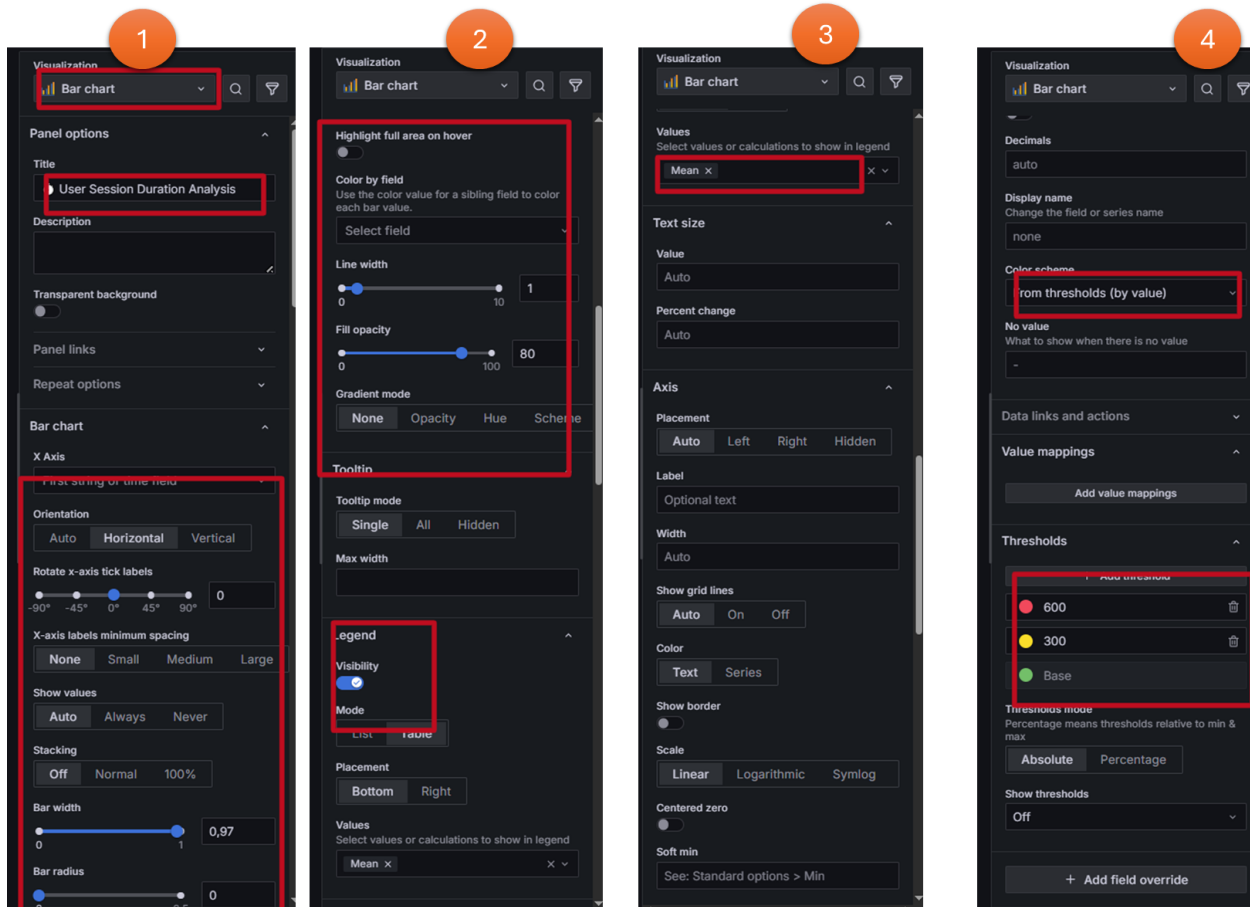
Panel: Bar Chart — "🕒 User Session Duration Analysis"





**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**

1. Add panel → Visualisation: **Bar chart**.
2. Query: `histogram_quantile(0.50, sum(rate(ebanking_session_duration_seconds_bucket{environment=~"$environment"}[10m])) by (le))`
3. Unit: `s` (seconds)
4. Options → Stacking: none (default), barWidth ~0.97.
5. Thresholds: warning 300s, critical 600s (configure in Field → Thresholds).
6. Title: 🕒 User Session Duration Analysis.



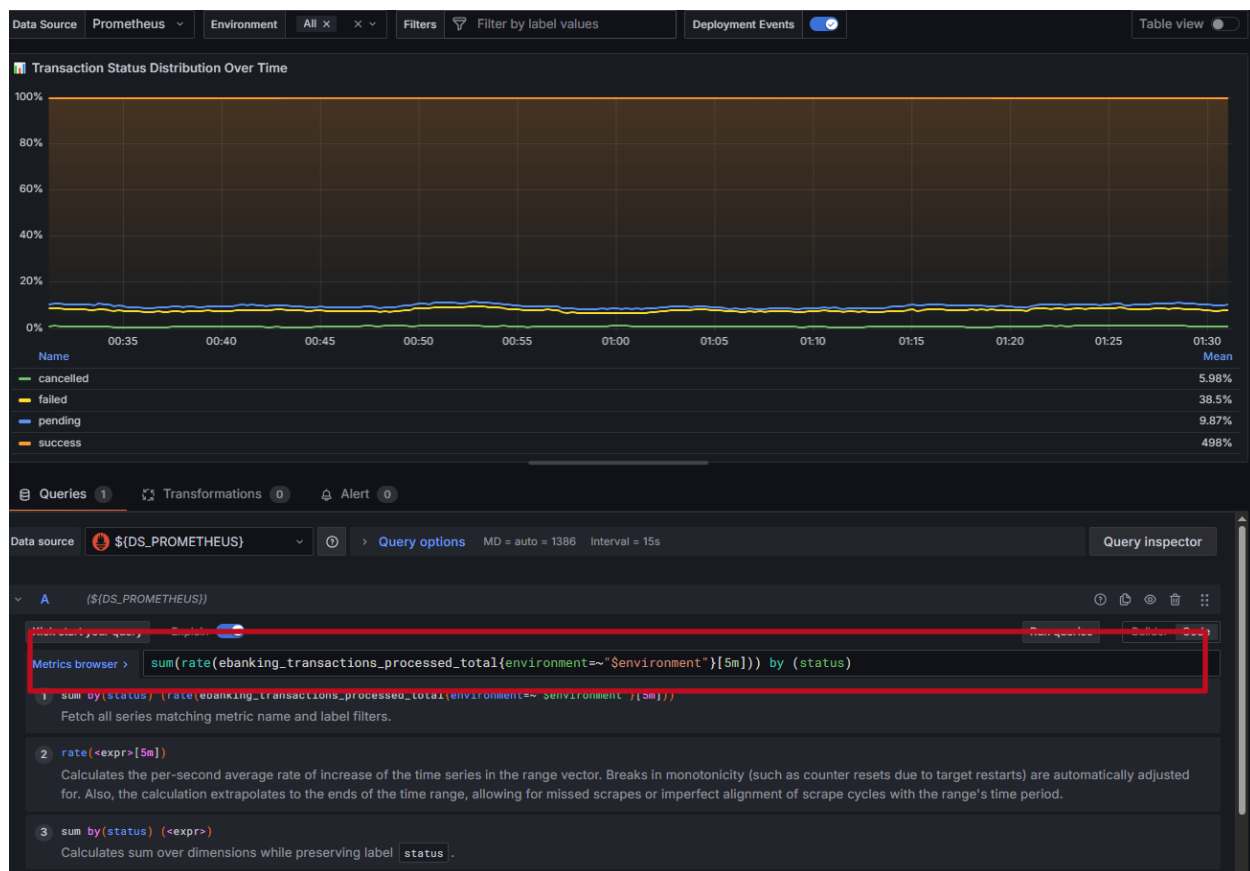
The image displays four sequential screenshots of the Grafana visualization configuration interface for a Bar chart, with specific settings highlighted by red boxes and numbered 1 through 4.

- Screenshot 1:** Shows the 'Panel options' section. The title is set to 'User Session Duration Analysis'. The 'Bar chart' section is expanded, showing 'X Axis' settings. The 'Orientation' is set to 'Horizontal'. The 'X-axis labels minimum spacing' is set to 'None'. The 'Show values' option is set to 'Auto'. The 'Stacking' is set to 'Off'. The 'Bar width' is set to 0.97.
- Screenshot 2:** Shows the 'Bar chart' section. The 'Highlight full area on hover' option is checked. The 'Color by field' section is expanded, showing 'Line width' set to 1 and 'Fill opacity' set to 80. The 'Gradient mode' is set to 'None'. The 'Tooltip mode' is set to 'Single'. The 'Legend' section is expanded, showing 'Visibility' checked and 'Mode' set to 'Table'.
- Screenshot 3:** Shows the 'Values' section. The 'Mean' value is selected. The 'Text size' is set to 'Auto'. The 'Value' is set to 'Auto'. The 'Percent change' is set to 'Auto'. The 'Axis' section is expanded, showing 'Placement' set to 'Auto'. The 'Label' is set to 'Optional text'. The 'Width' is set to 'Auto'. The 'Show grid lines' option is set to 'Auto'. The 'Color' is set to 'Text'. The 'Show border' option is checked. The 'Scale' is set to 'Linear'. The 'Centered zero' option is checked. The 'Soft min' is set to 'See: Standard options > Min'.
- Screenshot 4:** Shows the 'Thresholds' section. The 'Color scheme' is set to 'from thresholds (by value)'. The 'No value' is set to '-'. The 'Data links and actions' section is expanded. The 'Value mappings' section is expanded, showing 'Add value mappings'. The 'Thresholds' section is expanded, showing a table with thresholds: 600 (red), 300 (yellow), and Base (green). The 'Thresholds mode' is set to 'Absolute'. The 'Show thresholds' option is set to 'Off'.



**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**

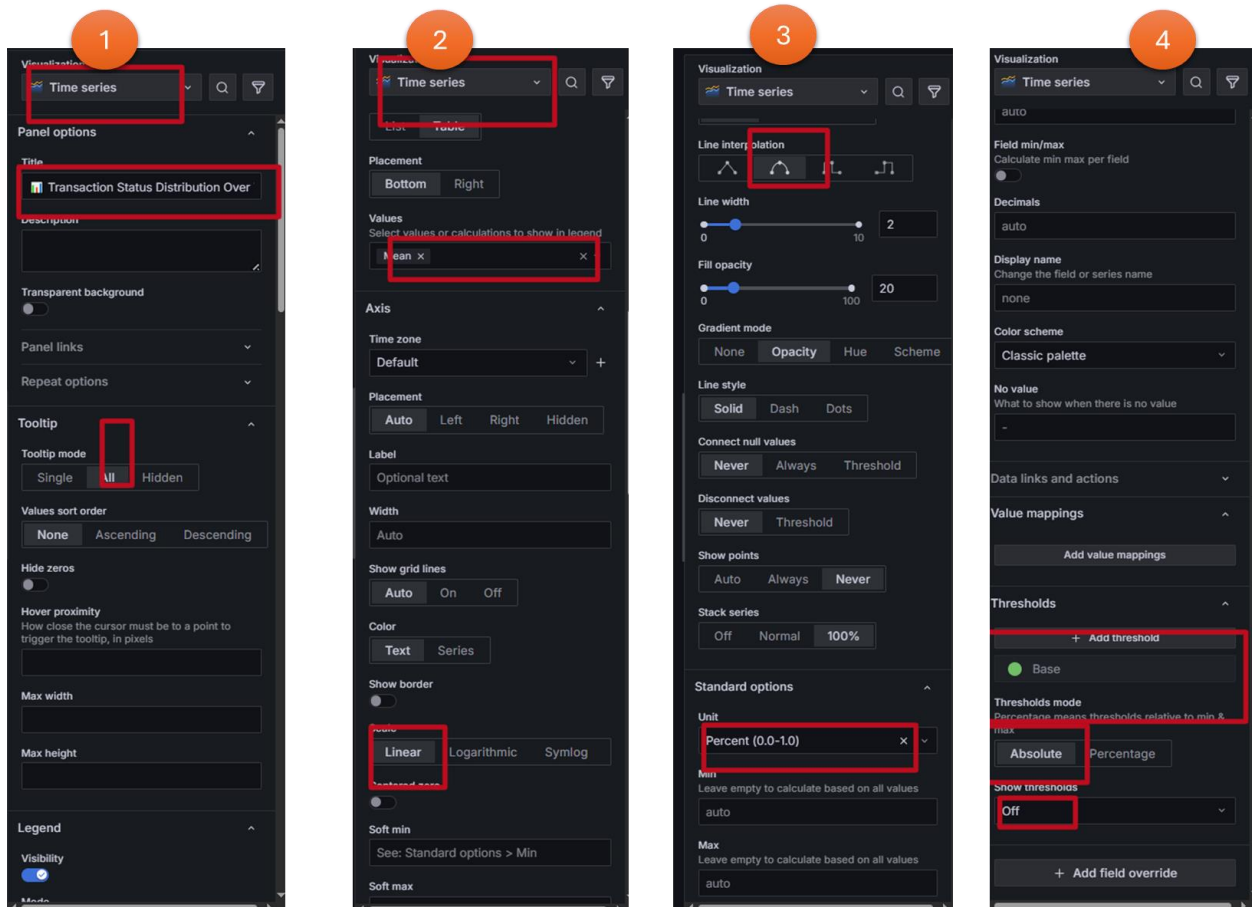
## Panel: Time Series — "Transaction Status Distribution Over Time"



1. Add panel → Time series.
2. Query:  
`sum(rate(ebanking_transactions_processed_total{environment=~'$environment'}[5m])) by (status)`
3. Visualization options: stacked area or stacked percent (set stacking mode to percent if you want % distribution).
4. Legend: use `{{status}}`.
5. Title: Transaction Status Distribution Over Time.



**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**

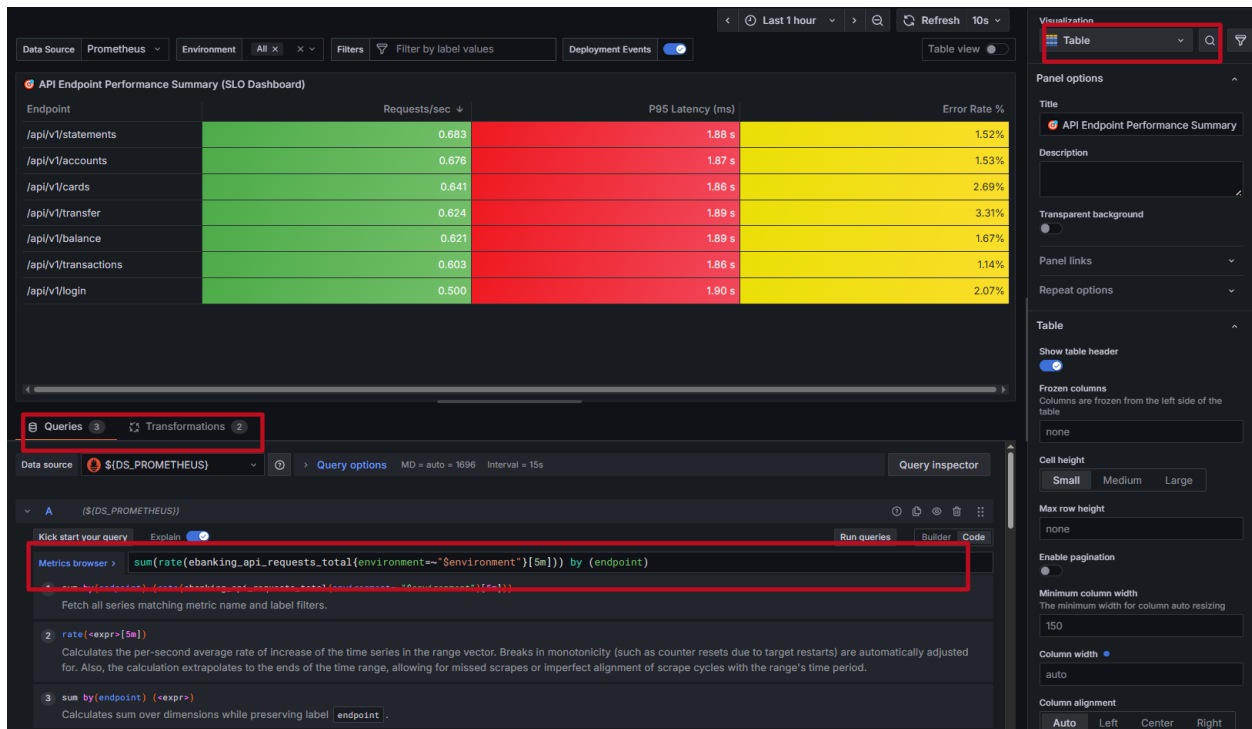


Panel: Table — "🎯 API Endpoint Performance Summary (SLO Dashboard)"

1. Add panel → Visualisation: **Table**.
2. Add three Targets (format = table, instant = true):



**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**



- A:  
`sum(rate(ebanking_api_requests_total{environment=~"$environment"}[5m])) by (endpoint) → format table (instant)`





**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**

- B: `histogram_quantile(0.95, sum(rate(ebanking_request_duration_seconds_bucket{environment=~"$environment"}[5m])) by (le, endpoint)) * 1000 → format table (instant)`

The screenshot shows the Grafana Metrics browser interface. The query editor contains the following query:

```
histogram_quantile(0.95, sum(rate(ebanking_request_duration_seconds_bucket{environment=~"$environment"}[5m])) by (le, endpoint)) * 1000
```

The query is broken down into five steps:

- `histogram_quantile(0.95, sum by (le, endpoint) (rate(ebanking_request_duration_seconds_bucket{environment=~"$environment"}[5m])) * 1000`  
Fetch all series matching metric name and label filters.
- `rate(<expr>[5m])`  
Calculates the per-second average rate of increase of the time series in the range vector. Breaks in monotonicity (such as counter resets due to target restarts) are automatically adjusted for. Also, the calculation extrapolates to the ends of the time range, allowing for missed scrapes or imperfect alignment of scrape cycles with the range's time period.
- `sum by (le, endpoint) (<expr>)`  
Calculates sum over dimensions while preserving labels `le` and `endpoint`.
- `histogram_quantile(0.95, <expr>)`  
Calculates the  $\phi$ -quantile ( $0 \leq \phi \leq 1$ ) from the buckets `b` of a histogram. The samples in `b` are the counts of observations in each bucket. Each sample must have a label `le` where the label value denotes the inclusive upper bound of the bucket. (Samples without such a label are silently ignored.) The histogram metric type automatically provides time series with the `_bucket` suffix and the appropriate labels.
- `<expr> * 1000`  
no docs

The 'Options' section at the bottom shows the following settings:

- Legend: ☒ Verbose
- Min step: auto
- Format:
- Type:

- 3.
- C: `(sum(rate(ebanking_api_requests_total{environment=~"$environment", status_code=~"5.."}[5m])) by (endpoint) / sum(rate(ebanking_api_requests_total{environment=~"$environment"}[5m])) by (endpoint)) * 100 → format table (instant)`



**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**

```

C ($DS_PROMETHEUS)
Kick start your query Explain
Run queries Builder Code

Metrics browser > (sum(rate(ebanking_api_requests_total{environment=~"$environment",status_code=~"5.."}[5m])) by (endpoint) / sum(rate(
ebanking_api_requests_total{environment=~"$environment"}[5m])) by (endpoint)) * 100


(sum by(endpoint), (rate(ebanking_api_requests_total{environment=~"$environment",status_code=~"5.."}[5m]) * 100) / (sum by(endpoint)
(rate(ebanking_api_requests_total{environment=~"$environment"}[5m]))))
Fetch all series matching metric name and label filters.

2 rate(<expr>[5m])
Calculates the per-second average rate of increase of the time series in the range vector. Breaks in monotonicity (such as counter resets due to target restarts) are automatically adjusted
for. Also, the calculation extrapolates to the ends of the time range, allowing for missed scrapes or imperfect alignment of scrape cycles with the range's time period.

3 sum by(endpoint) (<expr>)
Calculates sum over dimensions while preserving label endpoint.

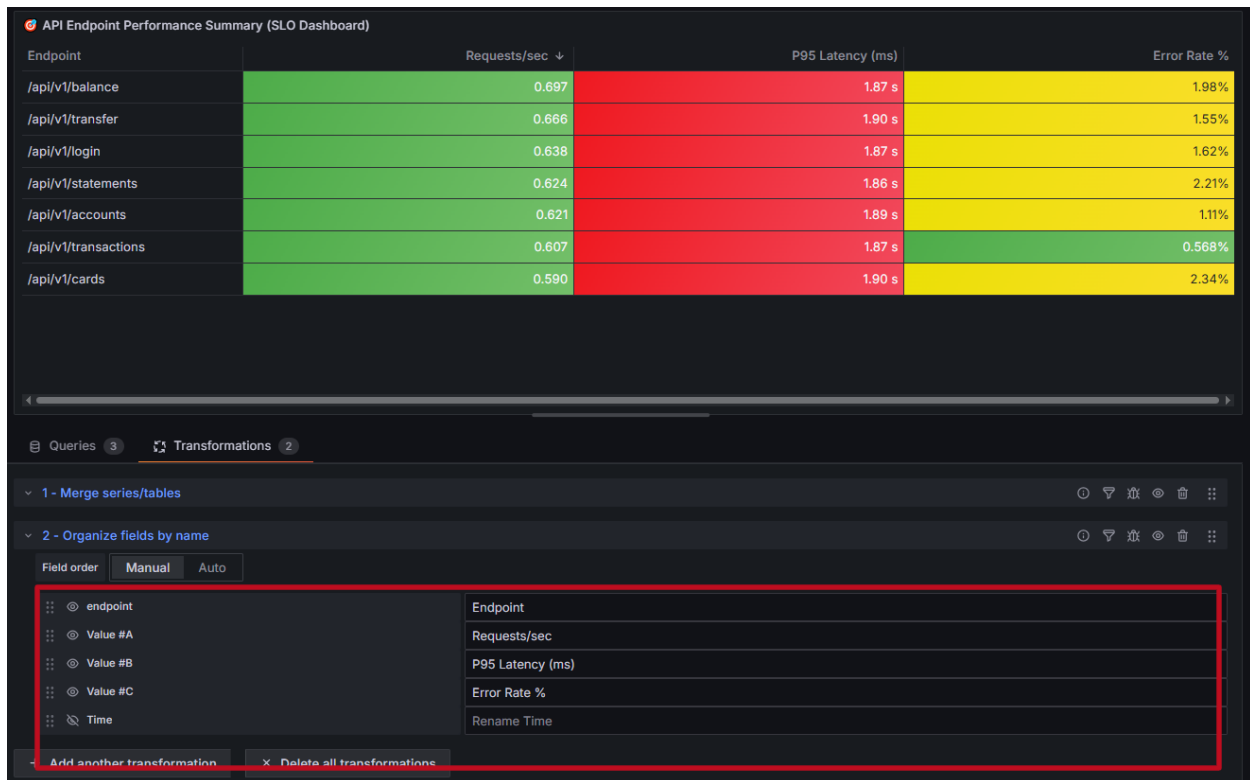
4 <expr> * 100
no docs

> Options Legend: Verbose Format: Table Step: auto Type: Instant
+ Add query + Expression
  
```

4. **Transformations** → **Merge (merge rows from A,B,C)** → **Organize fields: rename Value #A** → Requests/sec, Value #B → P95 Latency (ms), Value #C → Error Rate %, keep endpoint as Endpoint.
5. Field overrides: set P95 Latency (ms) unit ms, Error Rate % unit percent and color background thresholds (green/yellow/red) for quick scan.
6. Table options → Sort by Requests/sec desc.
7. Title:  API Endpoint Performance Summary (SLO Dashboard).



**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**



## 7) Row 4 — Training & Documentation

Panel: Text — " PromQL Learning Guide & Examples"

1. Add panel → Visualisation: **Text** (mode Markdown).
2. Copier-coller les snippets PromQL présents dans le JSON (débutant / intermédiaire / avancé).
3. Title: PromQL Learning Guide & Examples.





**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**

## 8) Annotations & Alerts via UI

### Ajouter une annotation "Deployment Events"

1. Grafana → Dashboard Settings → Annotations → Add Annotation Query
  - Name: Deployment Events
  - Data source: \${DS\_PROMETHEUS}
  - Expression:  
`changes(ebanking_app_info{environment=~"$environment"}[5m]) > 0`
  - Title: Deployment
  - Tag keys: version

### Créer des alertes simples (Grafana Alerting)

1. Grafana → Alerting → Alert Rules → New alert rule.
2. Exemple: High Error Rate
  - Query:  
`sum(rate(ebanking_api_errors_total{environment="$environment"}[5m])) / sum(rate(ebanking_api_requests_total{environment="$environment"}[5m])) * 100`
  - Condition: WHEN avg() OF query(A, 5m) IS ABOVE 5 → For 5 minutes.
  - Notifications: Slack / Email / PagerDuty channels.
3. Exemple: High P95 Latency
  - Query: `histogram_quantile(0.95, sum(rate(ebanking_request_duration_seconds_bucket{environment="$environment"}[5m])) by (le, endpoint)) * 1000`
  - Condition: > 1000 ms for 5m.

---

## 9) Tips graphiques et accessibilité

- Utiliser des emojis dans le titre des panels facilite la lecture rapide.
- Couleurs et seuils : privilégier palettes contrastées et signification métier.
- Légendes : activer `last` pour KPIs de synthèse, `mean` et `max` pour trends.





**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**

- Mode sombre vs clair : vérifier contraste pour les cellules colorées de la table.

## 10) Sauvegarde et export

- Dashboard → Share → Export → Save JSON to repo.
- Ajouter dans le repo `README.md` (voir proposition optionnelle).

## 11) Exercices guidés pas-à-pas (avec checks)

1. Importer le JSON et vérifier que tous les panels sont présents. (Check: 9 panels listés)
2. Re-cr  er manuellement le panel `Daily Revenue` et s'assurer que les thresholds s'affichent comme attendu.
3. Re-cr  er manuellement la table `SLO` et ex  cuter les transformations (merge + organize). V  rifier les unit  s.
4. Cr  er l'annotation `Deployment Events` et simuler une nouvelle version (via simulateur) — v  rifier qu'une annotation appar  t.
5. Cr  er la r  gle d'alerte `High Error Rate` et d  clencher en augmentant le taux d'erreur via le simulateur.

## 12) Assets & Placeholders pour la formation

- Screenshots recommand  s (placeholders) et points d'enregistrement :
  1. Datasource save & test screenshot
  2. Variable creation screenshot (environment)
  3. Panel create — Query editor screenshot
  4. Field/Thresholds pane screenshot
  5. Transformations table screenshot (merge + organize)
  6. Alert rule creation screenshot



**Formation: Grafana**  
**Formateur: Mokhtar Sellami**  
**Mail: [mokhtar.sellami@data2-ai.com](mailto:mokhtar.sellami@data2-ai.com)**

Inclure ces images dans le repo /images et les référencer dans le README.md.

---

## 13) Propositions de livrables que je peux générer

- README.md complet step-by-step (fichiers, commandes, variables, simulateur).
  - playbook.md pour l'inclusion du dashboard dans un pipeline CI (import via Grafana API).
  - screenshots.zip (placeholders ou réels si tu fournis accès/exports).
  - dashboard-export.json (une version adaptée si tu veux modifier les unités/seuils globalement).
- 

## Fin

Le document détaille chaque étape GUI, chaque champ utile du panneau et suggère les captures à inclure dans un kit de formation. Si tu veux, je peux maintenant :

- Générer README.md (instructions git + import JSON + simulateur).
- Générer un playbook pour l'importation via Grafana HTTP API (curl/JSON).
- Mettre à jour ce document avec screenshots réels si tu me fournis les images.

Dis-moi lequel tu veux et je le crée immédiatement dans le canvas.