

# Django Admin comme framework pour développer des outils internes



Romain Clement - PyConFR 2023

# Qui suis-je ?

## Romain Clement

Entrepreneur individuel

Ingénieur et formateur

Data | IA | DevOps

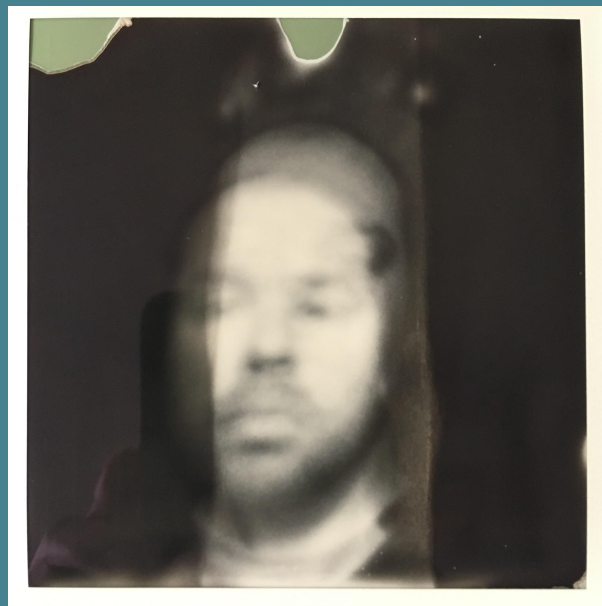
Organisateur Meetup Python Grenoble



[www.romain-clement.net](http://www.romain-clement.net)



[www.linkedin.com/in/romainclement/](https://www.linkedin.com/in/romainclement/)





Qui utilise le framework Django ?



Qui utilise l'application Django Admin ?



Qui développe des applications métier ?

 Cette présentation est pour vous !

# Introduction

*“Comment remplacer Excel par Django Admin ?”*

Mise en situation

Prototype

Retour d'expérience

---



Mise en situation



# Mise en situation - Contexte

Jean-Cérien Corp

Equipe développement des outils internes

Automatisation de processus métiers

# Mise en situation - Demande

*Le département performance souhaite évaluer le temps passé sur les différents projets de l'entreprise.*

*Jusqu'ici, les données étaient récoltées par chaque chef de projet dans des tableurs Excel et compilées ensemble en fin d'année.*

*Cela devient désormais ingérable, entre le nombre de collaborateurs (internes et externes), le nombre grandissant de projets, les données non formatées uniformément, les erreurs de saisie, je vous passe les détails.*

*Nous avons pris le pli de la "digitalisation" il y a maintenant une dizaine d'années, on nous promet une stratégie "data driven" mais on se retrouve à bidouiller des feuilles Excel ! S'il vous plaît, développez nous un outil interne pour automatiser ce workflow !*

***Jean-Louis, directeur performance***

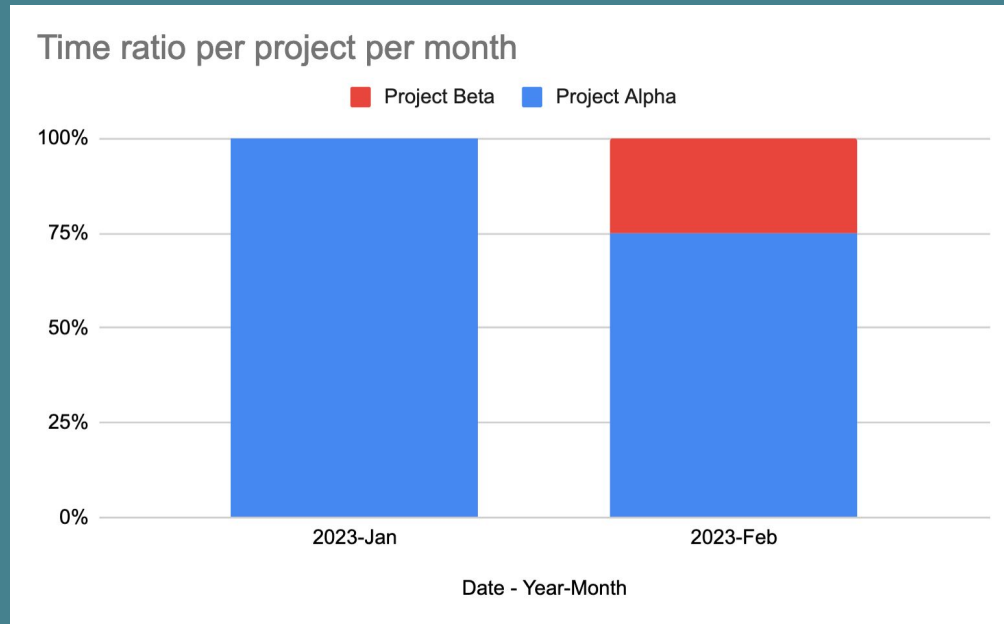
# Mise en situation - Données

	A	B	C	D	E
1	Date	Employee	Project	Duration	Comments
2	2023-02-02	Jean-Paul	Project Alpha	4:00:00	Debug
3	2023-02-02	Jean-Paul	Project Beta	4:00:00	Support
4	2023-02-02	Jean-Jacques	Project Alpha	8:00:00	Feature dev
5	2023-02-02	Jean-Michel	Project Alpha	8:00:00	Infra as code
6	2023-02-01	Jean-Paul	Project Beta	8:00:00	Debug
7	2023-02-01	Jean-Jacques	Project Alpha	8:00:00	Bugfix
8	2023-02-01	Jean-Michel	Project Alpha	8:00:00	Debug
9	2023-01-02	Jean-Jacques	Project Alpha	8:00:00	Bugfix
10	2023-01-02	Jean-Michel	Project Alpha	8:00:00	Feature dev

# Mise en situation - Métriques

	A	B	C	D
1	<i>SUM of Duration</i>	<i>Project</i>		
2	<i>Date - Year-Month</i>	Project Alpha	Project Beta	Grand Total
3	2023-Jan	16:00:00		16:00:00
4	2023-Feb	36:00:00	12:00:00	48:00:00
5	<b>Grand Total</b>	<b>52:00:00</b>	<b>12:00:00</b>	<b>64:00:00</b>

# Mise en situation - Graphes



# Mise en situation - Problèmes

	A	B	C	D	E
1	Date	Employé	Projet	Durée	Commentaires
2	2 février 2023	Jean-Paul C.	Projet alpha	4 heures	
3	2 février 2023	Jean Paul	Project Beta	8 heures	
4	02/02/23	Jean-Jacques B.	Project alpha	8 h	
5	02-02-2023	Jean-Michel	Projet Alpha	1 j	
6	1er février 2023	Jean-Paul	Project Beta	08h00	
7	01/02/23	Jean-Jacques	Project Alpha	08h	
8	01-02-2023	Jean-Michel	Project Alpha	1 jour	
9	02/01/23	Jean-Jacques	Project Alpha	8h	
10	02-01-2023	Jean-Michel	Project Alpha	8:00	

# Mise en situation - Objectifs

- Centralisation des données
- Uniformisation des données
- Calcul automatique des indicateurs
- Contrôle d'accès restreint

# Mise en situation - L'app dont vous êtes le héros

3 chemins possibles

Application sur-mesure



Outil no/low-code



Django Admin

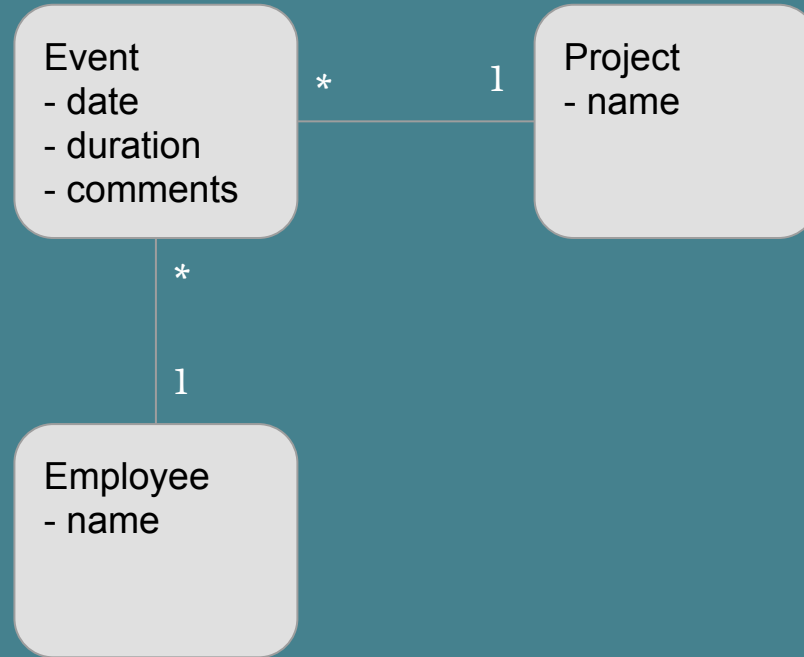




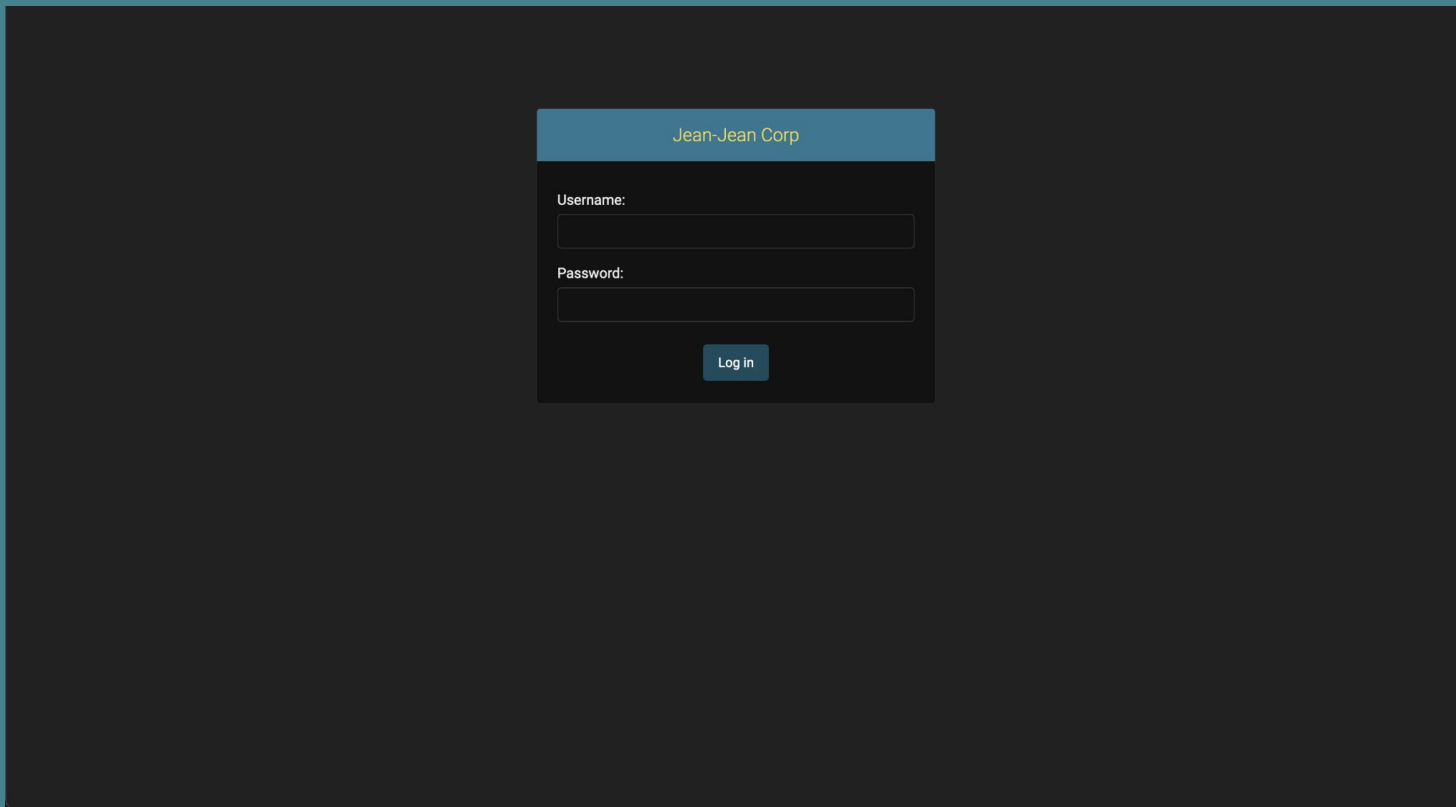


Prototype avec Django Admin !

# Prototype - Schéma de données



# Prototype - Démo



Jean-Jean Corp

Username:

Password:

Log in

# Prototype - Démo (staff)

Jean-Jean Corp

WELCOME, JEAN-PAUL. REPOSITORY / VIEW SITE / CHANGE PASSWORD / LOG OUT

Internal Management Tool

TIME TRACKING

Events [+ Add](#) [✎ Change](#)

Recent actions

My actions

- + 2023-02-16 - Jeanpaul - Project Beta  
- 1:00:00  
Event

# Prototype - Démo (staff)

Jean-Jean Corp WELCOME, JEAN-PAUL. REPOSITORY / VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Time Tracking › Events

Start typing to filter...

**TIME TRACKING**

Events + Add

### Select event to change

Search ADD EVENT +

◀ 2023 February 16 February 17

DATE	PROJECT	DURATION
Feb. 17, 2023	Project Alpha	8:00:00
Feb. 16, 2023	Project Beta	1:00:00

2 events

**FILTER**

↓ By project

All

Project Alpha

Project Beta

# Prototype - Démo (staff)

Jean-Jean Corp WELCOME, JEAN-PAUL. REPOSITORY / VIEW SITE / CHANGE PASSWORD / LOG OUT


Home › Time Tracking › Events › Add event

Start typing to filter...

**TIME TRACKING**

Events + Add

## Add event

**Date:**  Today |   
Note: You are 1 hour ahead of server time.

**Project:**  ▾

**Duration:**

**Comments:**

# Prototype - Démo (staff)

Jean-Jean Corp WELCOME, JEAN-PAUL. REPOSITORY / VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Time Tracking › Events › Add event

Start typing to filter...

**TIME TRACKING**

Events + Add

## Add event

**Please correct the error below.**

**Date:**  Today

Note: You are 1 hour ahead of server time.

**Project:**  ▾

**Duration:**  Enter a valid duration.

**Comments:**

# Prototype - Démo (admin)

Jean-Jean Corp

WELCOME, ROMAIN. REPOSITORY / VIEW SITE / CHANGE PASSWORD / LOG OUT

## Internal Management Tool

### ACCOUNTS

Users [+ Add](#) [Change](#)

### AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#) [Change](#)

### TIME TRACKING

Events [+ Add](#) [Change](#)

Project duration per month views [View](#)

Projects [+ Add](#) [Change](#)

## Recent actions

### My actions

[2023-02-16 - jeanmichel - Project Alpha - 8:00:00](#)  
Event

[+ Project Beta](#)  
Project

[+ 2023-02-16 - jeanmichel - Project Alpha - 8:00:00](#)  
Event

[+ Project Alpha](#)  
Project

[manager](#)  
User

[jeanpaul](#)  
User

[jeanmichel](#)  
User

[jeanmichel](#)  
User

[+ jeanmichel](#)  
User

[jeanpaul](#)  
User



# Prototype - Démo (admin)

The screenshot displays an admin interface for 'Jean-Jean Corp'. The top navigation bar includes the company name and user information: 'WELCOME, ROMAIN. REPOSITORY / VIEW SITE / CHANGE PASSWORD / LOG OUT'. The breadcrumb trail indicates the current location: 'Home > Time Tracking > Projects'.

The left sidebar contains a search bar and several menu categories:

- ACCOUNTS**
  - Users (+ Add)
- AUTHENTICATION AND AUTHORIZATION**
  - Groups (+ Add)
- TIME TRACKING**
  - Events (+ Add)
  - Project duration per month views
  - Projects (+ Add)**

The main content area is titled 'Select project to change' and features an 'ADD PROJECT +' button. Below the title, there is an 'Action:' dropdown menu, a 'Go' button, and a selection count '0 of 2 selected'. A list of projects is shown with checkboxes:

- PROJECT
- Project Beta
- Project Alpha

At the bottom of the list, it indicates '2 projects'.

# Prototype - Démo (admin)

Jean-Jean Corp WELCOME, ROMAIN. REPOSITORY / VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Time Tracking > Events

Start typing to filter...

ACCOUNTS

Users + Add

AUTHENTICATION AND AUTHORIZATION

Groups + Add

TIME TRACKING

Events + Add

Project duration per month views

Projects + Add

Select event to change

ADD EVENT + EXPORT

SEARCH

< 2023 February 16 February 17

Action: ▾ Go 0 of 3 selected

<input type="checkbox"/>	DATE	EMPLOYEE	PROJECT	DURATION
<input type="checkbox"/>	Feb. 17, 2023	jeanpaul	Project Alpha	8:00:00
<input type="checkbox"/>	Feb. 16, 2023	jeanpaul	Project Beta	1:00:00
<input type="checkbox"/>	Feb. 16, 2023	jeanmichel	Project Alpha	8:00:00

3 events

FILTER

↓ By project

All

Project Alpha

Project Beta

↓ By employee

All

jeanmichel

jeanpaul

manager

romain

# Prototype - Démo (admin)

Jean-Jean Corp WELCOME, ROMAIN. REPOSITORY / VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Time Tracking > Events > Add event

Start typing to filter...

**ACCOUNTS**

Users + Add

**AUTHENTICATION AND AUTHORIZATION**

Groups + Add

**TIME TRACKING**

Events + Add

Project duration per month views

Projects + Add

## Add event

**Date:**  Today |

Note: You are 1 hour ahead of server time.

**Employee:**

**Project:**

**Duration:**

**Comments:**

Save and add another Save and continue editing SAVE

# Prototype - Démo (admin)

Jean-Jean Corp WELCOME, ROMAIN. REPOSITORY / VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Time Tracking › Project duration per month views

Start typing to filter...

**ACCOUNTS**

Users + Add

**AUTHENTICATION AND AUTHORIZATION**

Groups + Add

**TIME TRACKING**

Events + Add

**Project duration per month views**

Projects + Add

### Select project duration per month view to view

YEAR	1 ▾	MONTH	2 ▾	PROJECT	TOTAL DURATION
2023		02		Project Beta	1:00:00
2023		02		Project Alpha	16:00:00

2 project duration per month views

**FILTER**

↓ By year

All

2023

↓ By month

All

02

↓ By project

All

Project Alpha

Project Beta

# Prototype - Démo (admin)

The screenshot displays an admin dashboard for 'Jean-Jean Corp'. The top navigation bar includes the company name and user information: 'WELCOME, ROMAIN. REPOSITORY / VIEW SITE / CHANGE PASSWORD / LOG OUT'. The breadcrumb trail indicates the current location: 'Home > Time Tracking > Events >'. A left sidebar menu is visible with categories: 'ACCOUNTS' (Users), 'AUTHENTICATION AND AUTHORIZATION' (Groups), 'TIME TRACKING' (Events, Project duration per month views, Projects), and a back arrow. The main content area is titled 'Export' and contains the text 'Export data to selected output format'. Below this, there is an 'Output:' label and a dropdown menu currently set to 'CSV'. A large, empty rectangular area is present below the dropdown, and a 'LAUNCH' button is located in the bottom right corner of this area.

Jean-Jean Corp

WELCOME, ROMAIN. REPOSITORY / VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Time Tracking > Events >

Start typing to filter...

ACCOUNTS

Users + Add

AUTHENTICATION AND AUTHORIZATION

Groups + Add

TIME TRACKING

Events + Add

Project duration per month views

Projects + Add

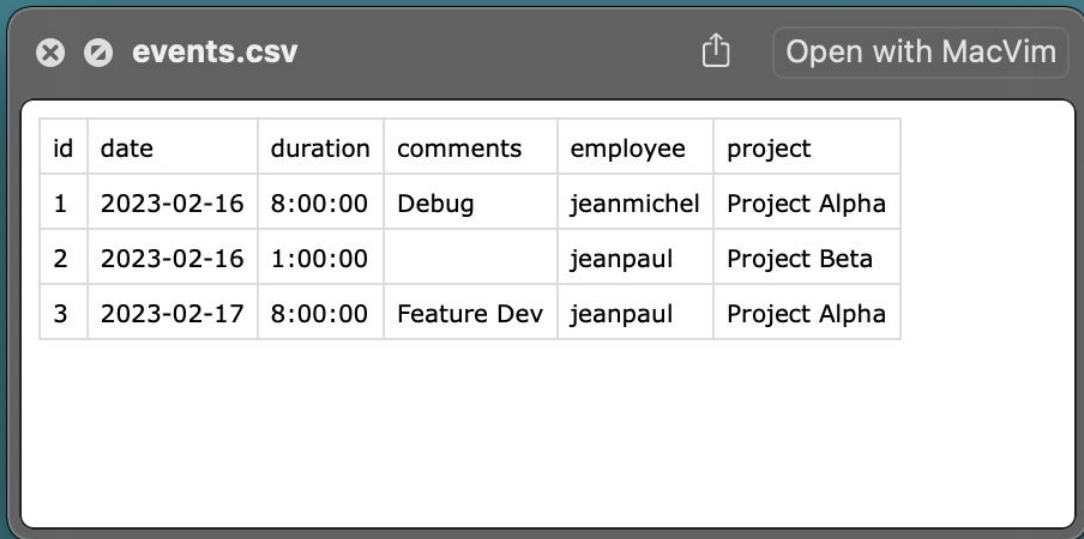
Export

Export data to selected output format

Output: CSV

LAUNCH

# Prototype - Démo (admin)



The image shows a window titled 'events.csv' with a 'Share' icon and a button that says 'Open with MacVim'. The window contains a table with the following data:

id	date	duration	comments	employee	project
1	2023-02-16	8:00:00	Debug	jeanmichel	Project Alpha
2	2023-02-16	1:00:00		jeanpaul	Project Beta
3	2023-02-17	8:00:00	Feature Dev	jeanpaul	Project Alpha



# Astuces pour Django Admin

# Prototype - Déclaration des modèles de données

```
class Project(models.Model):
    id = models.BigAutoField(verbose_name=_("id"), primary_key=True)
    name = models.CharField(
        verbose_name=_("name"), max_length=255, unique=True, null=False, blank=False
    )

    def __str__(self) -> str:
        return self.name

class Event(models.Model):
    id = models.BigAutoField(verbose_name=_("id"), primary_key=True)
    date = models.DateField(verbose_name=_("date"))
    duration = models.DurationField(verbose_name=_("duration"))
    comments = models.TextField(verbose_name=_("comments"), blank=True)
    employee = models.ForeignKey(
        settings.AUTH_USER_MODEL,
        verbose_name=_("employee"),
        on_delete=models.CASCADE,
        blank=False,
        null=False,
    )
    project = models.ForeignKey(
        Project,
        verbose_name=_("project"),
        on_delete=models.CASCADE,
        blank=False,
        null=False,
    )

    def __str__(self) -> str:
        return f"{self.date} - {self.employee} - {self.project} - {self.duration}"
```



# Prototype - Déclaration des vues

```
class EventAdmin(admin.ModelAdmin):  
    list_display = ("date", "employee", "project", "duration")  
    list_filter = ("employee", "project")  
    ordering = ("-date",)  
    date_hierarchy = "date"  
    search_fields = (  
        "employee__username",  
        "employee__first_name",  
        "employee__last_name",  
        "project__name",  
    )
```

# Prototype - Amélioration des formulaires

```
class EventAdmin(admin.ModelAdmin):  
    ...  
    fields = ("date", "employee", "project", "duration", "comments")
```

# Prototype - Vues agrégées

```
class ProjectDurationPerMonthView(models.Model):
    class Meta:
        managed = False

    id = models.BigAutoField(verbose_name=_("id"), primary_key=True)
    year = models.IntegerField(verbose_name=_("year"))
    month = models.IntegerField(verbose_name=_("month"))
    project = models.ForeignKey(
        Project,
        verbose_name=_("project"),
        on_delete=models.DO_NOTHING,
        blank=False,
        null=False,
    )
    total_duration = models.DurationField(verbose_name=_("total duration"))
```

# Prototype - Vues agrégées

```
operations = [  
  migrations.RunSQL(  
    sql="""  
        CREATE VIEW timetracking_projectdurationpermonthview AS  
        SELECT  
            row_number() OVER () AS id,  
            strftime('%Y', timetracking_event.date) AS year,  
            strftime('%m', timetracking_event.date) AS month,  
            timetracking_event.project_id AS project_id,  
            sum(timetracking_event.duration) AS total_duration  
        FROM timetracking_event  
        GROUP BY year, month, project_id;  
    """,  
    reverse_sql="""  
        DROP VIEW timetracking_projectdurationpermonthview;  
    """,  
  ),  
  migrations.CreateModel(  
    name="ProjectDurationPerMonthView",
```

# Prototype - Vues agrégées

```
class ProjectDurationPerMonthViewAdmin(admin.ModelAdmin):
    list_display = ("year", "month", "project", "total_duration")
    list_display_links = None
    list_filter = ("year", "month", "project")
    ordering = ("-year", "-month")
    sortable_by = ("year", "month", "project", "total_duration")

    def has_module_permission(self, request: HttpRequest) -> bool:
        | return request.user.is_superuser or request.user.is_staff

    def has_view_permission(
    | self, request: HttpRequest, obj: Model | None = None
    ) -> bool:
        | return request.user.is_superuser or request.user.is_staff

    def has_add_permission(self, request: HttpRequest) -> bool:
        | return False

    def has_change_permission(
    | self, request: HttpRequest, obj: Model | None = None
    ) -> bool:
        | return False

    def has_delete_permission(
    | self, request: HttpRequest, obj: Model | None = None
    ) -> bool:
        | return False
```

# Prototype - Contrôle d'accès

```
def has_module_permission(self, request: HttpRequest) -> bool:
    return request.user.is_superuser or request.user.is_staff

def has_view_permission(
    self, request: HttpRequest, obj: Model | None = None
) -> bool:
    return request.user.is_superuser or request.user.is_staff

def has_add_permission(self, request: HttpRequest) -> bool:
    return request.user.is_superuser or request.user.is_staff

def has_change_permission(
    self, request: HttpRequest, obj: Model | None = None
) -> bool:
    return request.user.is_superuser or request.user.is_staff

def has_delete_permission(
    self, request: HttpRequest, obj: Model | None = None
) -> bool:
    return request.user.is_superuser
```

# Prototype - Contrôle d'accès

```
def get_queryset(self, request: HttpRequest) -> QuerySet[Any]:
    qs = super().get_queryset(request)
    if not request.user.is_superuser:
        qs = qs.filter(employee=request.user)
    return qs

def get_list_display(self, request: HttpRequest) -> Sequence[str]:
    list_display = ["date", "project", "duration"]
    if request.user.is_superuser:
        list_display.insert(1, "employee")
    return list_display

def get_list_filter(self, request: HttpRequest) -> Sequence[str]:
    list_filter = ["project"]
    if request.user.is_superuser:
        list_filter.append("employee")
    return list_filter

def get_fields(
    self, request: HttpRequest, obj: Model | None = ...
) -> Sequence[Union[Callable[..., Any], str]]:
    fields = ["date", "project", "duration", "comments"]
    if request.user.is_superuser:
        fields.insert(1, "employee")
    return fields
```

# Prototype - Pages et formulaires

```
def get_urls(self) -> list[URLPattern]:
    urls = super().get_urls()
    info = self.model._meta.app_label, self.model._meta.model_name
    custom_urls = [
        path(
            "export/",
            self.admin_site.admin_view(self.export),
            name="%s_%s_export" % info,
        ),
    ]
    return custom_urls + urls
```



# Prototype - Pages et formulaires

```
def export(self, request: HttpRequest, extra_context: Any = None) -> HttpResponse:
    if not request.user.is_superuser:
        self.message_user(
            request,
            _("You are not authorized to access this page"),
            level=messages.ERROR,
        )
        return redirect("../")

    if request.method == "POST":
        form = ExportForm(request.POST)
        if form.is_valid():
            ...
    else:
        form = ExportForm()

    context = {
        **self.admin_site.each_context(request),
        "title": _("Export"),
        "description": _("Export data to selected output format"),
        "opts": self.model._meta,
        "form": form,
        **(extra_context or {}),
    }

    return render(
        request,
        "admin/timetracking/event/export.html",
        context,
    )
```

```
class ExportForm(forms.Form):
    class OutputChoice(TextChoices):
        CSV = "csv", _("CSV")

    output = forms.ChoiceField(
        label=_("Output"), choices=OutputChoice.choices, initial=OutputChoice.CSV
    )

    required_css_class = "required"
```

# Prototype - Pages et formulaires

```
{% extends "admin/change_form.html" %}
{% load i18n admin_urls static admin_modify %}

{% block content %}
<div id="content-main">
  <p>{{ description }}</p>

  <form method="POST" enctype="multipart/form-data">
    {% if form.errors %}
      <p class="errornote">
        {% if form.errors.items|length == 1 %}
          {% translate "Please correct the error below." %}
        {% else %}
          {% translate "Please correct the errors below." %}
        {% endif %}
      </p>
    {% endif %}

    {% csrf_token %}

    <fieldset class="module aligned wide">
      {% for field in form %}
        <div class="form-row">
          {{ field.errors }}
          {{ field.label_tag }} {{ field }}
          {% if field.help_text %}
            <p class="help">{{ field.help_text|safe }}</p>
          {% endif %}
        </div>
      {% endfor %}
    </fieldset>

    <div class="submit-row">
      <input type="submit" value="{% translate 'Launch' %}" class="default">
    </div>
  </form>
</div>
{% endblock %}
```

# Prototype - Pages et formulaires

```
{% extends 'admin/change_list_object_tools.html' %}
{% load i18n admin_urls %}

{% block object-tools-items %}
    {{ block.super }}
    {% if user.is_active and user.is_superuser %}
        <li>
            <a href="{% url cl.opts|admin_urlname:'export' %}" class="historylink">
                {% translate "export" %}
            </a>
        </li>
    {% endif %}
{% endblock %}
```

# Prototype - Pour aller plus loin

Quelques exemples de possibilités

- Single Sign-On : `djanto-allauth`
- Intégrer d'autres applications : `django-revproxy`
- Tâches asynchrones : `django-q`, `celery`, `dagster`
- Import / export : `django-import-export`
- Historique et undo / redo : `django-simple-history`
- Graphes : `chartjs`, `vega-lite`
- Test : `pytest-django`
- Typage statique : `django-stubs`



**Retour d'expérience**

# Retour d'expérience - Points positifs

- Compromis entre low-code et application sur-mesure
- Approche déclarative (données, vues)
- Autogénération du frontend web
- Customisation des vues
- Possibilité de sortir de Django Admin
- Intégration dans les pipelines de CI/CD
- Déploiement agnostique et multi-environnements
- Contrôle des données
- FLOSS

# Retour d'expérience - Points négatifs

- Customisation parfois complexe
- Compréhension intrinsèque nécessaire
- Forte dépendance aux modèles
- Peu de ressources sur les usages avancés

# Ressources

- [Django Admin Reference](#)
- [Django Admin Cookbook](#)
- [Haki Benita Django Admin Articles](#)

Illustrations : Wikimedia Commons, Romain Clement



# Merci de votre attention !

 Questions ?



 [www.romain-clement.net](http://www.romain-clement.net)

 [www.linkedin.com/in/romainclement/](https://www.linkedin.com/in/romainclement/)

 [rclement/pyconfr2023-djangoadmin](https://rclement.pyconfr2023-djangoadmin)

---