

Spécification Technique Détaillée Générateur de Bulletin de Paie

Proof of Concept - Projet de Paie Industrialisé

21 août 2025

Table des matières

1	Architecture du Système	2
2	Étape 1 : Chargement des Données	2
3	Étape 2 : Calcul de la Rémunération Brute	2
3.1	Salaire de base et Heures Supplémentaires	2
3.2	Avantage en Nature (Logement)	2
3.3	Salaire Brut Total	3
4	Étape 3 : Calcul de la Réduction Générale	3
4.1	Ajustement du SMIC de référence	3
4.2	Calcul du Coefficient (C) et de la Réduction	3
5	Étape 4 : Calcul des Cotisations Sociales	3
6	Étape 5 : Calcul du Salaire Net et du Net Versé	4
6.1	Du Brut au Net à Payer	4
6.2	Calcul des Indemnités de Frais Professionnels	4
6.3	Du Net à Payer au Net Versé	4
A	Structure des Fichiers de Configuration	4
A.1	parametres_entreprise.json	4
A.2	parametres_contrat.json	5
A.3	taux_cotisations.json	5
A.4	baremes.json	5

1 Architecture du Système

Le système est conçu de manière modulaire pour séparer clairement les données, la logique de calcul et les outils de mise à jour.

- **Configuration (dossier config/)** : Fichiers JSON contenant toutes les données.
- **Scripts de mise à jour (dossier scripts/)** : Outils Python (scrapers) chargés de mettre à jour la configuration depuis des sources officielles.
- **Moteur de calcul (generateur_fiche_paie.py)** : Le script principal qui orchestre le calcul et l’affichage.

Flux de travail

1. L'utilisateur exécute les scripts du dossier `/scripts` pour mettre à jour les barèmes et les taux dans les fichiers JSON.
2. L'utilisateur renseigne les données de l'entreprise, du contrat et les variables du mois dans les fichiers JSON.
3. L'utilisateur exécute `generateur_fiche_paie.py` qui lit toutes les configurations et produit le bulletin de paie.

2 Étape 1 : Chargement des Données

Le processus débute par le chargement en mémoire de l'ensemble des fichiers de configuration.

```
1 # Dans generateur_fiche_paie.py
2 try:
3     with open('config/taux_cotisations.json', 'r') as f:
4         taux_cotisations = json.load(f)['TAUX_COTISATIONS']
5     with open('config/parametres_contrat.json', 'r') as f:
6         contrat = json.load(f)['PARAMETRES_CONTRAT']
7     with open('config/parametres_entreprise.json', 'r') as f:
8         entreprise = json.load(f)['PARAMETRES_ENTREPRISE']
9     with open('config/baremes.json', 'r') as f:
10         baremes = json.load(f)
11 except Exception as e:
12     print(f"Erreur de chargement : {e}")
13     return
```

3 Étape 2 : Calcul de la Rémunération Brute

La rémunération brute est la somme de tous les éléments de salaire soumis à cotisations.

3.1 Salaire de base et Heures Supplémentaires

$$\begin{aligned}\text{SalaireBase} &= \text{TauxHoraire} \times \text{HoraireMensuel} \\ \text{MontantHS} &= \text{QtéHS} \times (\text{TauxHoraire} \times 1.25)\end{aligned}$$

3.2 Avantage en Nature (Logement)

La valeur de l'avantage est déterminée en consultant le barème chargé depuis `baremes.json`. Le script recherche la première tranche de rémunération supérieure au salaire de référence du salarié.

```

1 montant_avantage_logement = 0
2 nb_pieces = contrat['variables_du_mois'].get('avantage_logement_nb_pieces', 0)
3 if nb_pieces > 0:
4     bareme_logement = baremes['AVANTAGES_EN_NATURE_2025']['logement_bareme_forfaitaire']
5     for tranche in bareme_logement:
6         if salaire_base <= tranche['remuneration_max']:
7             montant_avantage_logement = tranche['valeur_1_piece'] if nb_pieces
8             == 1 else tranche['valeur_par_piece'] * nb_pieces
9             break

```

3.3 Salaire Brut Total

C'est la somme des éléments positifs, de laquelle on soustrait les absences non rémunérées.

$$\text{SalaireBrut} = (\text{SalaireBase} + \dots + \text{AN}) - \text{DédutionAbsence}$$

4 Étape 3 : Calcul de la Réduction Générale

Le calcul suit la formule officielle mensualisée de l'URSSAF.

4.1 Ajustement du SMIC de référence

Le SMIC est ajusté en cas de temps partiel ou d'absence non rémunérée.

$$\begin{aligned} \text{SMIC}_{\text{base}} &= \text{SMIC}_{\text{Horaire}} \times \frac{35 \times 52}{12} \\ \text{SMIC}_{\text{proratisé}} &= \text{SMIC}_{\text{base}} \times \frac{\text{DuréeHebdo}}{35} \\ \text{SMIC}_{\text{ajusté}} &= \text{SMIC}_{\text{proratisé}} \times \frac{\text{SalaireBrutRéal}}{\text{SalaireBrutHabituel}} \end{aligned}$$

4.2 Calcul du Coefficient (C) et de la Réduction

$$C = \min \left(\left(\frac{T}{0.6} \right) \times \left(1.6 \times \frac{\text{SMIC}_{\text{ajusté}}}{\text{SalaireBrut}} - 1 \right), T \right)$$

$$\text{RéductionGénérale} = -(C \times \text{SalaireBrut})$$

```

1 parametre_T = entreprise['constantes_paie_2025']['parametre_T_reduction_generale']
2 reduction_generale = 0
3 if salaire_brut > 0 and salaire_brut < (1.6 * smic_ajuste):
4     coefficient_C = (parametre_T / 0.6) * (1.6 * smic_ajuste / salaire_brut - 1)
5     coefficient_C = min(coefficient_C, parametre_T)
6     reduction_generale = - (coefficient_C * salaire_brut)

```

5 Étape 4 : Calcul des Cotisations Sociales

Le script itère sur chaque cotisation, détermine la base de calcul et applique les taux.

```

1 for code, info in taux_cotisations.items():
2     if info.get('base') == 'brut': base_calcul = salaire_brut
3     elif info.get('base') == 'plafond_ss': base_calcul = min(salaire_brut,
4     # ... autres bases ...
5

```

```

6   montant_salarial = base_calcul * info.get('salarial', 0) + info.get('
salarial_fixe', 0)
7   montant_patronal = base_calcul * info.get('patronal', 0) + info.get('
patronal_fixe', 0)

```

6 Étape 5 : Calcul du Salaire Net et du Net Versé

6.1 Du Brut au Net à Payer

$$\begin{aligned}
 \text{NetAvantImpôt} &= \text{SalaireBrut} - \sum \text{CotisationsSalariales} \\
 \text{NetImposable} &= \text{NetAvantImpôt} + \text{CSG/CRDS}_{\text{non déd.}} + \text{PartPatronaleMutuelle} \\
 \text{Impôt} &= \text{NetImposable} \times \text{TauxPAS} \\
 \text{NetÀPayer} &= \text{NetAvantImpôt} - \text{Impôt}
 \end{aligned}$$

6.2 Calcul des Indemnités de Frais Professionnels

Les indemnités non soumises à cotisations sont calculées à partir des barèmes.

```

1 # Exemple pour le petit d placement
2 km_deplacement = contrat['variables_du_mois'].get('frais_petit_deplacement_km',
0)
3 if km_deplacement > 0:
4     for tranche in baremes['petit_deplacement_bareme']:
5         if tranche['km_min'] <= km_deplacement < tranche['km_max']:
6             montant = tranche['montant']
7             break
8     if contrat['variables_du_mois'].get('vehicule_electrique', False):
9         montant *= 1.20 # Majoration de 20%

```

6.3 Du Net à Payer au Net Versé

$$\text{NetVersé} = \text{NetÀPayer} + \sum \text{IndemnitésFraisPro} - \text{MontantSaisi}$$

```

1 total_indemnites = indemnite_repas + indemnite_deplacement + ...
2 net_verse = net_a_payer + total_indemnites - montant_saisi

```

A Structure des Fichiers de Configuration

A.1 parametres_entreprise.json

```

1 {
2   "PARAMETRES_ENTREPRISE": {
3     "identite": { "raison_sociale": "", "siret": "", ... },
4     "conditions_cotisations": { "effectif_total": 0, ... },
5     "constantes_paie_2025": { "smic_horaire": 0, ... }
6   }
7 }

```

A.2 parametres_contrat.json

```
1 {  
2   "PARAMETRES_CONTRAT": {  
3     "identite": { "nom": "", "prenom": "", ... },  
4     "poste": { "emploi": "", "date_entree": "YYYY-MM-DD", ... },  
5     "remuneration": { "taux_horaire": 0, ... },  
6     "variables_du_mois": { "heures_supplementaires_majoration_25": 0, ... },  
7     "imposition": { "taux_pas": 0, ... },  
8     "compteurs": { "conges_solde_anterieur": 0, ... }  
9   }  
10 }
```

A.3 taux_cotisations.json

```
1 {  
2   "TAUX_COTISATIONS": {  
3     "securite_sociale_maladie": {  
4       "libelle": "...", "base": "...", "salarial": 0, "patronal": 0  
5     }, ...  
6   }  
7 }
```

A.4 baremes.json

```
1 {  
2   "FRAIS_PROFESSIONNELS_2025": {  
3     "repas_indemnites": {},  
4     "petit_deplacement_bareme": [],  
5     ...  
6   },  
7   "SAISIE_SUR_SALAIRE_2025": { ... },  
8   "AVANTAGES_EN_NATURE_2025": { ... }  
9 }
```