

Document de consignes — Structure et contenu d'un CV

Objectif : rédiger le contenu d'un CV professionnel, lisible et optimisé pour les recruteurs techniques (développeur Java).

1. Principes généraux

- **Lisibilité avant tout.** Police lisible, taille 10–12 pt pour le corps, 14–18 pt pour les titres. Espaces suffisants entre les sections.
 - **Clarté & hiérarchie.** Utiliser titres, sous-titres et listes à puces. Le recruteur doit pouvoir scanner le CV en 6–8 secondes.
 - **Pertinence.** Adapter le contenu au poste ciblé : mettre en avant technologies et expériences pertinentes.
 - **Brevité.** 1 page pour un junior / 1–2 pages maximum pour profils avec plusieurs années d'expérience.
 - **Vérifiabilité.** Ne pas exagérer : préférer des verbes d'action précis et des résultats mesurables si possible.
 - **Cohérence visuelle.** Limiter les polices (max 2), palette de couleurs limitée (2–3 couleurs).
-

2. Ordre recommandé des sections (format développeur junior)

1. **En-tête / Titre** (nom + rôle) — ligne supérieure
2. **Accroche / Profil professionnel** (1–3 phrases)
3. **Compétences techniques (hard skills)** — listes ou pictos
4. **Expériences professionnelles** (chronologie inversée)
5. **Projets / Réalisations** (liens GitHub, déploiements)
6. **Formations & certifications**
7. **Soft skills et langues**
8. **Informations complémentaires** (présence en ligne : GitHub / LinkedIn / portfolio) / centres d'intérêt

Adapter l'ordre si nécessaire : ex. placer les compétences en tête si le candidat a des compétences techniques fortes mais peu d'expérience.

3. Détail des sections et modèles de rédaction

En-tête

- **Nom** (gras, plus grand), **titre** (ex. Développeur Java Junior)
- Coordonnées : téléphone, email professionnel, ville (p.ex. Lyon, France), lien LinkedIn, GitHub, site perso (si pertinent).

- Exemple : `Romain Dupont – Développeur Java Junior | +33 6 00 00 00 00 | romain.dupont@mail.com | github.com/romaindupont`

Accroche / Profil professionnel (3 lignes max)

- Résumez qui vous êtes, votre valeur ajoutée et votre objectif.
- Exemple : « Développeur Java junior, passionné par les architectures backend et les microservices. Expérience sur Spring Boot et tests automatisés. Recherche un poste backend pour contribuer à des plateformes évolutives. »

Compétences techniques

- Organiser par catégories : **Langages, Frameworks & Bibliothèques, Base de données, Outils & CI/CD, Tests.**
- Indiquer un niveau simple si nécessaire (Ex. *Confirmé / Intermédiaire / Notions*), ou utiliser barres/pictos pour le visuel.
- Exemple :
 - **Langages** : Java (SE/EE), Kotlin (notions)
 - **Frameworks** : Spring Boot, Spring MVC, Hibernate
 - **BDD** : PostgreSQL, MySQL
 - **Outils** : Git, Docker, Maven

Expériences professionnelles

- Ordre : titre du poste — entreprise — période (mois/année).
- Description : 3–6 bullet points par expérience, axés sur réalisations (verbes d'action + résultats quantifiables si possible).
- Format : `Tâche / Contexte – Action – Résultat`.
- Exemple :
 - **Développeur Java — Startup X — 06/2023–08/2024**
 - Participation à la conception d'une API REST (Spring Boot) consommée par 20k utilisateurs mensuels.
 - Mise en place de pipelines CI/CD (GitHub Actions), réduction du temps de déploiement de 40%.

Projets / Réalisations

- Inclure 2–4 projets significatifs (projets perso, contributions open-source, projets d'études).
- Pour chaque projet : titre, technologies, description courte (objectif + votre rôle), lien (GitHub, démo).
- Exemple : `Plateforme de gestion de tâches – Java, Spring Boot, PostgreSQL – API REST, authentification JWT – https://github.com/...`

Formations & certifications

- Diplôme — Établissement — année.
- Certifications (ex. Oracle Java Certification, cours en ligne pertinents).

Soft skills & langues

- Soft skills : travail en équipe, autonomie, communication, résolution de problèmes.
- Langues : indiquer niveau (B2, C1 ou *courant / notions*).

Informations complémentaires

- Liens : GitHub, LinkedIn, site personnel, portfolio.
 - Disponibilité : préavis, mobilité (si demandé).
 - Centres d'intérêt (1 ligne max) — choisir des éléments pouvant enrichir un entretien (ex. contribution à la communauté OSS, meetups tech).
-

4. Titres et sous-titres — exemples prêts à l'emploi

- **Accroche** : Profil — Accroche — À propos de moi
 - **Compétences** : Compétences techniques — Tech stack — Outils
 - **Expériences** : Expériences professionnelles — Parcours — Missions
 - **Projets** : Projets sélectionnés — Réalisations techniques
 - **Formation** : Formation & Certifications
 - **Autres** : Langues — Centres d'intérêt — Liens utiles
-

5. Conseils rédactionnels pratiques

- Utiliser des **verbes d'action** : Conçu, Développé, Optimisé, Automatisé, Corrigé.
 - Préférer le format **bullet points** (clairs et courts). 6 lignes max par expérience.
 - Pas de phrases longues — phrases nominales possibles.
 - Vérifier orthographe et syntaxes ; demander une relecture externe.
 - Adapter le vocabulaire au poste (ex. « microservices », « REST », « tests unitaires » si pertinent).
-

Fiche mémo — Adobe Express (outils, bonnes pratiques, export)

Objectif : aide-mémoire complet et opérationnel sur l'usage d'Adobe Express pour créer un CV : interface, outils essentiels, bonnes pratiques visuelles et paramètres d'export.

1. Présentation rapide d'Adobe Express

- **Qu'est-ce que c'est ?** Un outil en ligne de création graphique orienté templates et rapidité (équivalent orienté template par rapport à des outils plus complexes).
 - **Avantages** : modèles prêts à l'emploi, interface simple, intégration d'images/icônes, export PDF/JPG/PNG, génération de QR code.
 - **Usage pour CV** : rapide, accessible, idéal pour créer un CV visuellement cohérent sans compétences avancées en design.
-

2. Interface – éléments principaux

- **Tableau de bord / Mes projets** : accès aux créations précédentes.
 - **Templates / Modèles** : recherche par mot-clé ("CV", "résumé", "resume").
 - **Barre d'outils (gauche ou supérieure selon version)** : Texte, Images, Icones/Elements, Arrière-plan, Couleurs, Effets, Téléchargement / Export.
 - **Zone de travail (canvas)** : zone centrale où éditer.
 - **Propriétés du calque** : dimension, alignement, espacement, opacité.
-

3. Étapes rapides pour créer un CV dans Express

1. **Choisir un modèle** proche du rendu souhaité (sobriété/tech). Chercher par "CV tech" ou "resume".
 2. **Importer votre contenu** : copier-coller l'en-tête, accroche, compétences, expériences.
 3. **Organiser les blocs** : grouper sections, ajuster marges et colonnes.
 4. **Remplacer images/icônes** : utiliser la bibliothèque d'icônes (chercher "Java", "code", "GitHub"), ou importer vos propres pictogrammes.
 5. **Ajuster typographie** : choisir 1 police pour titres, 1 police pour corps (contraste clair). Respecter la hiérarchie (h1, h2, body).
 6. **Couleurs** : appliquer palette 2-3 couleurs. Vérifier contraste pour l'accessibilité.
 7. **Finaliser** : vérifier alignements, orthographe, liens cliquables.
 8. **Exporter** : PDF haute qualité (voir section export).
-

4. Outils et éléments utiles dans Adobe Express

- **Blocs de texte** : titres, sous-titres, corps. Styles prédéfinis.
 - **Grilles & colonnes** : organiser le CV en 1, 2 ou 3 colonnes.
 - **Icones / pictogrammes** : rechercher par mot-clé (ex. "code", "github", "email").
 - **Images** : importer photo pro ou illustrations (formats PNG/JPG). Préférer photo carrée, recadrée.
 - **Formes & séparateurs** : barres, lignes pour structurer visuellement.
 - **Couleurs & palettes** : créer palette personnalisée ou utiliser suggestions.
 - **Alignement & espacement** : fonctions d'alignement automatique, grille magnétique.
 - **Calques** : envoyer vers l'arrière / avant.
 - **QR code** : génération intégrée pour lier GitHub/portfolio.
-

5. Raccourcis & bonnes pratiques (générales, valables dans la majorité des versions)

- **Copier-coller** texte depuis un éditeur (préférer .txt ou Google Docs pour éviter styles parasites).
 - **Utiliser des repères** : activer les guides pour aligner précisément.
 - **Grouper** : grouper éléments pour les déplacer ensemble.
 - **Verrouiller** les blocs une fois positionnés pour éviter de les déplacer accidentellement.
 - **Prévisualiser** en taille réelle avant l'export.
-

6. Conseils typographiques et couleurs

- **Polices** : Sans serif lisible (ex. Inter, Open Sans) pour le corps ; une police plus marquante pour le titre si souhaité.
 - **Hiérarchie** : Titre (18–24 pt), Sous-titres (12–16 pt), Corps (10–12 pt).
 - **Contrastes** : texte sombre sur fond clair ou inversement ; s'assurer que le contraste respecte la lisibilité.
 - **Icônes** : taille cohérente, style homogène (ne pas mélanger icônes plates et détaillées).
-

7. Exportation et formats

- **PDF haute qualité (recommandé)** : pour envoi par email / candidature en ligne — conserver les liens cliquables (si possible). Nom du fichier : `CV_PrenomNom_YYYY.pdf`.
 - **PDF pour impression** : vérifier marges et fonds (si fond coloré, prévoir saignées si impression pro).
 - **PNG/JPG** : pour sites web ou portfolio (résolution 300 dpi conseillée pour l'impression, 72–150 dpi pour web).
 - **Taille & poids** : optimiser images (compressions) pour que le PDF reste léger (< 2–3 MB si possible).
 - **Vérifier les liens cliquables** (LinkedIn, GitHub) avant d'envoyer.
-

8. Astuces spécifiques pour candidatures techniques

- Mettre les **liens cliquables** vers GitHub, démos, contributions ; ajouter un QR code vers le profil ou un projet important.
 - Si vous avez des *benchmarks* (ex. nombre d'utilisateurs, fréquence de déploiement), les citer brièvement.
 - Ajouter une **section Projets** visible si peu d'expérience pro mais projets significatifs.
-

Checklist « CV pro »

But : une liste de vérification finale, pratique, à utiliser avant export et envoi d'un CV.

Contrôle du fond (contenu)

- [] Titre clair (ex. Développeur Java) et accroche présente.
- [] Compétences techniques correctement catégorisées.
- [] Expériences listées en ordre chronologique inverse.
- [] Chaque expérience contient 2–6 bullet points axés sur réalisations.
- [] Projets ou contributions open-source mentionnés avec liens.
- [] Formations et certifications notées (avec date).
- [] Langues & soft skills indiqués.
- [] Aucun mensonge ni exagération.

Contrôle de la forme (design)

- [] Une police principale pour le corps, une pour les titres (max 2).
- [] Taille et hiérarchie typographique lisibles.
- [] Espacements réguliers et marges suffisantes.
- [] Alignements cohérents (gauche / centré selon modèle).
- [] Palette de couleurs limitée (2-3 couleurs) et contrastée.
- [] Icônes/pictos cohérents et utiles (pas trop décoratifs).
- [] Photo professionnelle (facultative) respecte proportions et style.

Contrôle technique (Adobe Express / export)

- [] Tous les liens (LinkedIn, GitHub, portfolio) fonctionnent.
- [] Liens cliquables testés dans le PDF exporté.
- [] QR code si ajouté pointe vers la bonne URL.
- [] Export en PDF haute qualité : nommage correct (CV_PrenomNom_YYYY . pdf).
- [] Taille du fichier raisonnable (< 2-3 MB si possible).
- [] Vérifier une impression test (si le CV sera imprimé).

Vérification finale (orthographe & tonalité)

- [] Relecture orthographique automatisée + relecture humaine.
- [] Les formulations sont concises et orientées action.
- [] Le ton est adapté au poste (professionnel, technique).
- [] Aucun anglicisme superflu ; vocabulaire technique exact.

Préparation à l'envoi

- [] Nommer correctement le fichier et créer une version PDF et une version texte si nécessaire.
- [] Si envoi par mail : préparer un objet clair (ex. « Candidature — Développeur Java — Romain Dupont »).
- [] Préparer une version LinkedIn / PDF optimisée (format plus lisible sur mobile si nécessaire).
- [] Sauvegarder une copie modifiable (dans Adobe Express) pour futures mises à jour.

Annexes utiles (modèles de phrases et exemples rapides)

Phrases d'accroche possibles

- « Développeur Java junior spécialisé en backend et microservices, expérimenté dans Spring Boot et CI/CD. »
- « Ingénieur logiciel orienté backend, passionné par la performance et la qualité du code. »

Verbes d'action fréquents

- Conçu, Développé, Implémenté, Optimisé, Corrigé, Automatisé, Documenté, Déployé.

Format date et périodicité

- Utiliser ou selon clarté, ex. .
-