Comment faire passer un entretien technique ?

Boîte à outils et tips

[Introduction](#_pz4cuaoquyp8)

[Préparation de l’entretien](#_twxcysumq1i8)

[Déroulé de l’entretien](#_jmkutgqzpnp1)

[ICE BREAKER](#_9ixgbswfbk3r)

[INTRODUCTION](#_tkq4fjc93d52)

[PRÉSENTATION DU CANDIDAT](#_n17jjvywd563)

[KATA - live programing](#_aqp2clloro4r)

[Exercice du “Jeu de Tennis” à faire en TDD](#_a6lfg7u3xvsk)

[Alternative : Exercice de refacto - kata Gilded Rose](#_3p76itsh5t1u)

[Clôture de l'exercice](#_dyw7x7i9cgg8)

[LE CANDIDAT PRÉSENTE SON CODE](#_o68lcz9gkwzh)

[QUESTIONS/RÉPONSES](#_480hwfgmaoa5)

[WRAP UP / Projection sur la suite](#_268f5ime2c16)

[Après l’entretien](#_4zj6vc2pvbuh)

[BONUS : spécial stagiaire](#_hcsqbb55gv9k)

[Présentation du sujet de stage](#_iwohcclq5w4r)

[Tester l'intérêt du candidat pour le sujet de stage](#_uc52ckwiguu9)

[Tester son autonomie](#_iuhx1uvlv4p6)

[BONUS : étude de cas (pour les CC+ NAD)](#_5ytszoo6azfm)

[BONUS : exercices orientés “data”](#_pl9hlh3ngfln)

[Exercice SQL](#_c8fab12jfx3e)

[Questions pour mesurer les notions en BDDR](#_vk23tfwxtojl)

[Exercice de modélisation](#_iivvsfasvxp)

[BONUS : exercices orientés “architecture” et “DevOps”](#_6n7uurph5sja)

[Etude de cas](#_xbbg46ldxvfa)

[Phases](#_3890l0h4vjh9)

# 

# Introduction

Cette « boîte à outils » répertorie :

* Ce que l’on souhaite tester au cours d’un entretien et les outils pour y parvenir
* Définition d’un format type d’entretien avec les principaux exercices
* Des tips

=> Sorte de package « clé en main » pour aller en entretien (presque) les mains dans les poches

A qui ça s’adresse :

* Personnes qui n’ont jamais fait passer d’entretiens / jamais pour OCTO, qui ont peur de se lancer ou ne savent pas comment s’y prendre
* Les ceintures noires de l’entretien, pour partager nos méthodes

BTW Pourquoi faire passer des entretiens ?

* Apprentissage, échanges enrichissants
* Choisir ses futurs collègues !
* « Dédramatisation » des entretiens pour le futur (lorsque vous passerez de l’autre côté du bureau)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Préparation de l’entretien

1. **Étude du CV** du candidat + de sa fiche sur TWIG.
2. **De quelles informations ai-je besoin** pour décider si c'est un bon candidat ?

En fonction du niveau du candidat, mes exigences ne sont pas les mêmes :

* CC : notamment essayer de mesurer sa capacité à prendre le rôle de tech lead + Posture conseil
* Profil junior : tester s’il a des bases solides et s’il a la capacité à monter en compétences rapidement / progresser rapidement sur une nouvelle expertise
* Stagiaire : tester son autonomie et son intérêt pour le sujet de stage.

Quel que soit le niveau, il est primordial de

Dans tous les cas, la principale question à laquelle je dois répondre est :

* **Est-ce que j’aimerais partir en mission avec cette personne ?**
* **Est-ce que je me projette de travailler avec lui dans mon équipe ?**

1. Ainsi définir quels exercices / questions **piocher dans la boîte à outils.**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Déroulé de l’entretien

## ICE BREAKER

Discussion informelle pour mettre le candidat à l’aise.

Le classique qui marche pas en remote : “ça va pas trop compliqué de venir ? Vous habitez loin ?”

Acter le tutoiement en expliquant que c’est la norme chez OCTO.

## INTRODUCTION

Durée : ~5 min

1. Présentation des recruteurs.
2. Présentation du processus de recrutement (l’entretien tech fait suite à un entretien RH, et sera suivi d’un entretien manager)
3. Présentation du format de l’entretien (durée, plusieurs ateliers…)
4. Confirmer avec le candidat sa disponibilité
   1. *Tip : prévoir 1h30 d’entretien et évoquer la possibilité de déborder (soit de suite, soit au bout d'une heure)*
   2. *Prévoir de s’arrêter 15 minutes avant la fin de l’entretien afin de laisser au candidat l’occasion de poser des questions à des vrais consultants*
5. Partage de nos attentes
   1. L’entretien est un échange
   2. Le candidat doit nous convaincre sur ses capacités techniques (relative à son niveau d’expé)
   3. Nous devons lui donner envie de rejoindre OCTO

## PRÉSENTATION DU CANDIDAT

Durée : ~15 min

*Idées de questions sur son parcours / ses expériences, à approfondir en fonction du candidat.*

* **Rôle et intégration dans une équipe** 
  + Rôle ?
  + Tech lead ?Évaluer la définition du rôle de tech lead
  + Travail en équipe ou en solo ?
* **Méthodologies utilisées**
  + Projet en Agile ?

Évaluer le niveau de connaissance des valeurs et principes de l’agile

* **« solution-oriented » ou « technology-oriented »**
  + Poser des questions sur le métier afin de mesurer son intérêt pour le métier
* **Compétences techniques et pratiques de développement**
  + Choisir une ou deux technos parmi celles écrites sur le CV et demander au candidat de les présenter
  + Projet mis en PROD ?
  + Utilisation de TDD ?
  + Pratique de code review ?
  + Intérêt pour l'architecture ?
  + Intérêt pour l’OPS ?
  + Utilisation de Git ?
* **Envie d'apprendre, de monter en compétences**
  + Lectures ?
  + Veille ?
  + MOOC ou formation suivi récemment ?
  + Participation à des meetup ?
* **Profil “NAD”** (en fonction du parcours du candidat)
  + Pourquoi le passage de la BI vers le Big Data ?
  + Projection : Data Engineer ou Data Scientist ?
* **Ambition**
  + Quelles sont ses ambitions ?
  + Ce qu'il a envie de faire chez OCTO ?
  + Pourquoi Octo ?

## KATA - live programing

Durée : ~30-45 min

**Ce que l’on va tester** :

* Compétences techniques et pratiques de développement
  + Aisance avec IDE
  + Utilisation de raccourcis
  + Connaissance de la programmation
  + Notions de qualité de code
  + etc
* Fonctionnement
  + Réaction aux feedbacks
  + Écoute
  + Questionnement
  + Ose dire je ne sais pas
  + Force de propositions
  + Réflexions
  + Communication
  + Demander d'aide en cas de blocage
  + etc

*Note : On n'attend pas d'un candidat junior qu’il ait réponse à tout ou qu’il soit expert du langage. On veut s’assurer qu’il ait des notions de base et qu’il a le potentiel de maîtriser rapidement de nouvelles technos.*

**Tips - Outils pour un entretien tech à distance**

* Simple partage d’écran
* Visual Studio + plugin [Live Share](https://marketplace.visualstudio.com/items?itemName=MS-vsliveshare.vsliveshare-pack)
* IntelliJ Idea + plugin [teamHub](https://plugins.jetbrains.com/plugin/11955-teamhub) (inconvénient : besoin d’un partage d’écran pour voir les commandes lancer dans le terminal du candidat en particulier pour les tests)

**Présentation des modalités de l’exercice:**

1. Kata = exercice de code
2. Demander au candidat sur quel langage il est à l’aise et **choisir le langage** de l’exercice en conséquence (évite que le candidat perde ses moyens car il ne connaît pas le langage).
3. Présentation des “règles du jeu”
   1. C’est le candidat qui lead, les recruteurs sont là pour le guider et l’aiguiller si besoin.
   2. Nous sommes conscient que c’est intimidant de coder devant plusieurs personnes, rassurer le candidat, qu’il n’hésite pas à tester des choses
   3. Si un doute (sur syntaxe ou autre), possible d’aller sur google
   4. À priori on ira pas jusqu’au bout de l’exercice, pas d’inquiétude
4. Objectif de l’exercice : voir aisance du candidat à coder et s’assurer qu’il a de bonnes notions

### Exercice du “Jeu de Tennis” à faire en TDD

* **Présentation de l’exercice**
  + Les joueurs commencent avec un score de 0.
  + Les joueurs marquent des points. (rappel score : 0 -> 15 -> 30 -> 40)
  + Objectif : afficher le score du match de Tennis
    - exemples : “0 - 0”, “15 - 30”
    - Affiche “deuce” en cas d’égalité à 40-40
* **Présentation du TDD**
  + Poser un test qui échoue
  + Modifier le code pour faire passer le test
  + Refacto : changer la structure du code, sans changer son comportement

Lorsque l’on ajoute un test, il faut que tous les autres tests continuent de passer

Objectif: avoir un design émergent, une implémentation la plus simple et naturelle pour répondre au besoin métier

* **Demander au candidat les cas de test**
  + quand un jeu commence, le score est de 0-0
  + Si joueur 1 marque, le score est de 15 - 0
  + Si joueur 2 marque, le score est de 0 -15
  + Si joueur 1 marque deux fois, le score est de 30 - 0
  + Si joueur 1 marque trois fois, le score est de 40 - 0
  + Si joueur 1 marque une fois et le joueur 2 marque une fois, le score est de 15 - 15
  + ...
* **Poser ensemble le premier test**
  + Avec Given, When, Then
* **Le candidat prend le lead**

Note : possible de fournir comme base au candidat un projet avec le set-up pour pouvoir lancer des tests unitaires rapidement. En Java, le candidat devra cloner le repo suivant : <https://github.com/Akireken/TennisGame.git> (peut être fait en amont de l’entretien)

### Alternative : Exercice de refacto - kata [Gilded Rose](https://github.com/emilybache/GildedRose-Refactoring-Kata)

* **Présentation du** [**kata**](https://github.com/emilybache/GildedRose-Refactoring-Kata/blob/master/GildedRoseRequirements_fr.md)
* **Accès au code**

1. **Demander au candidat son avis sur le code existant**
   1. Lisibilité ?
   2. Qualité ?
   3. Des choses qui le choquent ?
   4. Est-ce qu’il se sent prêt à ajouter la nouvelle fonctionnalité ?
   5. Qu’est ce qu’il changerait ?

Ce qui peut être constaté :

* Legacy
* Difficile de lire et de comprendre le code
* Bad smells (une unique fonction très longue, nombreuses conditions, répétitions, magic number...)
* Quasiment aucun test / mauvaise couverture de code
* Risque de régression en essayant d’ajouter la nouvelle fonctionnalité
* **Comment ajouter la nouvelle fonctionnalité ?**
  1. Quelle stratégie ?
  2. Qu’est-ce qui devrait être refactorer en premier ?

**1) Comprendre l’application existante en lisant le code**

Intuition : se raccrocher au métier et le comprendre pour être guidé.

⚠ Trop complexe : approche contre-productive

**Tips** : oublier pour un temps le métier et faire l’assomption que le code de production a raison

**2) Lancer le test existant.** Le test fail.

Corriger, renommer, puis consolider le test existant (notamment ajouter condition sur Sellin et Quality)

**4) Ajouter des tests**

Utiliser un outil de code coverage

Objectif : couvrir au maximum de code (en un minimum d’efforts)

**5) Refactorer :** changer la structure sans changer le comportement

Maintenant qu’on dispose d’un filet de sécurité, on peut refactorer le code en toute quiétude

* **Passer à la pratique**

### Clôture de l'exercice

Faire un court debrief :

* Demander au candidat ce qu’il en a pensé, s’il aurait fait les choses différemment, si ça lui a paru difficile…
* Faire un feedback

## LE CANDIDAT PRÉSENTE SON CODE

Sorte de “live code review”

**Ce que l’on va tester** :

* Compétences techniques et pratiques de développement
* Fonctionnement
  + Capacité à prendre du recul
  + Capacité à donner son avis
  + Réaction aux feedback
  + ...

**Questions à poser au candidat** :

* Pourquoi il est fier (ou non) de ce code ?
* Pourquoi il a fait certains choix techniques ?
* Ce qu’il aurait pu faire de mieux ?
* Ce qu’il changerait avec le recul ?

Alternative : Revue d’un code “neutre” (par exemple récupéré au hasard sur github) dans un langage connu par le candidat

## QUESTIONS/RÉPONSES

Ces questions peuvent être posées pendant l’exercice de code.

**Ce que l’on va tester** :

* Notions du candidat
* Réaction si on sort de son expertise
* Ose-t’il dire qu’il ne sait pas

**Notions de POO**

* Qu’est-ce qu’un constructeur ?
* Qu’est-ce qu’un objet ?
* Quelle est la différence entre une classe et un objet ?
* Quelle est la différence entre attribut et héritage ?
* Qu’est-ce que l’héritage ?
* Qu’est-ce que le polymorphisme ?
* Qu’est-ce qu’une classe abstraite ?
* …

**Notions de tests et de leur utilité**

* Décrire les différentes typologies de tests connues ainsi que leur objectif
  + unitaire
  + intégration
  + fonctionnel
  + end-to-end
* Notions sur la pyramide de tests d’un projet ?
  + Durée d’exécution
  + Feedback
  + ...

**Culture G domaine IT** :

* Qu’est ce qu’un DNS?
* Qu’est ce qu’un load balancer ?
* Qu’est ce qu’un reverse proxy ?
* As-tu des notions de cloud ?
* ...

**Spécifiques à l’expertise du candidat → exemple pour un futur NAD**

* Quelles grandes familles de BDD connais-tu ?
* Quels types de BDD NoSQL connais-tu ?
* ...

## WRAP UP / Projection sur la suite

Durée : ~10 min

1. Demander au candidat s’il a des questions.
2. Demander au candidat son ressenti sur l’entretien

Tips : à ce moment de l’entretien, il est possible de s'absenter avec le binôme recruteur pour faire un débrief : vérifier que nous sommes alignés sur le feedback ainsi que sur la décision GO/NO GO, éventuellement s’il reste un doute déterminer une façon d’interroger le candidat...

1. Feedback au candidat sur ce que l’on a pensé de l’entretien.

Mentionner les points positifs de l'entretien et les points d'amélioration

Être factuel

Objectif : qu’il puisse repartir avec des choses actionnables

1. Faut-il annoncer la décision (si c’est un GO / NO GO) ?

Plusieurs écoles sur ce sujet.

Ma stratégie :

* Si vous n'êtes pas à l'aise pour dire que c'est un NO GO ou pas complètement sûre du GO, simplement dire au candidat que les RHs reviendront vers lui.
* Si c'est un GO sûre et sans équivoque, l'annoncer à l'issue de l'entretien permet d'éviter au candidat de patienter pour un retour, stresser etc.

1. Présentation de la suite du process de recrutement

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Après l’entretien

1. Remplir la fiche Twig du candidat
2. Faire un retour aux RHs (objectif : améliorer ensemble le processus de recrutement)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# BONUS : spécial stagiaire

## Présentation du sujet de stage

Une partie de l'entretien doit être dédiée à la présentation du sujet de stage.

## Tester l'intérêt du candidat pour le sujet de stage

* Pourquoi ce stage et pas un autre ?
* Qu’est-ce que tu as compris de ce sujet de stage ?
* Tester les connaissances du candidat sur le sujet de stage pour déterminer :
  + ses connaissances initiales
  + s’il a fait des recherches en amont de l’entretien pour creuser le sujet
  + s’il se projette dans le stage et à identifier comment il pourrait traiter le sujet

## Tester son autonomie

*Ça me semble crucial pour un stagiaire OCTO qui travaille la majeure partie du temps seul et qui doit “construire” son sujet.*

Questions sur ses précédents stages / précédentes expériences professionnelles :

* Comment s’organisait-il dans son quotidien ?
* Son tuteur était-il très présent ? Ou au contraire était-il très autonome ?
* Comment faisait-il s'il y avait un blocage ? Qui sollicitait-il ?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# BONUS : étude de cas (pour les CC+ NAD)

Durée : ~30-45 min

Exercice : [Banque COQ](https://docs.google.com/document/d/1ife4xDHgquVcF-gAVtmFnAtai6UNMhrQZj-NatcaUKs/edit?usp=sharing) (catégorisation de mouvements bancaires et exposition pour différentes apps)

**En amont de l’entretien**:

* Fournir l’étude de cas au candidat. Une semaine à l’avance pour qu’il ait le temps de la préparer.
* Préciser :
  + Le candidat pourra présenter l’architecture sur le support de son choix (dessin en live sur tableau blanc ou via MIRO, ppt, etc)
  + Le candidat peut poser des questions par mail

Avant de commencer, **présentation du cadre de l’exercice au candidat** :

* Les recruteurs se mettent dans la position du client.
* Il n’y a pas “une” archi ou “une” solution.
* Objectif : échanger.

A tester :

* Forme :
  + Qualité du support de la présentation
  + Clarté de la présentation
* Métier :
  + Présentation du contexte avant de présenter la solution ?
  + Prise en compte du contexte métier dans le choix de la solution ?
  + Est-ce que la solution proposée correspond au besoin métier ?
  + Propositions fonctionnelles ?
* Technique :
  + Sur quoi se concentre le candidat (réponse plutôt orientée ML ou mise en place de Spark) ?
  + Collecte des données : questionner l'utilisation de batch ou streaming ?
  + Stockage des données : où ? comment ? sous quel format ?
  + Où stocker les données pour l’exposition ?
  + Traitement des données : spark ?
  + ETL ou ELT ?
  + Partage des données avec les autres entités : Pub/Sub? Donner accès en lecture à un database? comment gérer les accès?
  + Solution Cloud vs on-premise?
  + Fraicheur des données ?
  + BDD indexée ? Quel index ?
* Stratégie de mise en place de l’architecture :
  + Présentation de la trajectoire ?
  + Par quel composant commencer ?
  + Paralléliser ?
  + Quelle équipe pour la réalisation ?
* Tips :
  + En cas de “Bluff” : si le candidat parlent de nombreuses techno et survole, creuser une techno en particulier
  + Éviter trop de “je ne sais pas” → il doit savoir réfléchir et trouver des réponses si on lui donne des pistes

Proposition d’architecture :

1. "collect" : couche d’import des différents types de données et standardisation
2. "modélisation": transformation et stockage dans le bon format
3. "service": exposition des données

Archi inspirée de PFM (data hub BNP)

1. Récupérer les fichiers et les mettre sur HDP
2. Nettoyage pour récupérer uniquement certains mouvements
3. Fichiers propres utilisés par Spark
4. Déduction de la catégorie : via règles métier et pour les autres via prédiction (ML)
5. Enregistrement dans Cassandra

→ Orchestration via Oozie

Alternative :

1. Nettoyage
2. Écriture dans HDP
3. Utilisation par Spark

Pour aller plus loin : <https://octo.atlassian.net/wiki/spaces/RH/pages/10387624/Pour+tous+niveaux+d+exp+rience+les+outils>

Note : autre étude de cas : Apple Bank → étude de cas plus généraliste

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# BONUS : exercices orientés “data”

## **Exercice SQL**

* Tester les bases du SQL : faire écrire quelques requêtes simples
  + SELECT … FROM … WHERE …
  + Sélectionner que certaines colonnes, ou certaines lignes
  + Faire un COUNT
  + Requête avec un GROUP BY
  + Utiliser un alias
  + Faire une jointure

## Questions pour mesurer les notions en BDDR

* Quelle est la différence entre une table et une vue ?
* Qu’est-ce qu’une clé primaire ? une clé étrangère ?
* Définir la cardinalité
* Qu’est-ce qu’un ORM ?

## Exercice de modélisation

Demander au candidat de construire un modèle de données, en indiquant les clés primaires/étrangères, les cardinalités, comment faire si j'ai une relation many-to-many? ...

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# BONUS : exercices orientés “architecture” et “DevOps”

## Etude de cas

[Cas SPOVED (DevOps)](https://docs.google.com/document/d/1B15VOO_TCCvVa0_VDd5_Szmnfq_j7p-qVgDXtGqoDRQ/edit)

[Cas Applebank](https://docs.google.com/document/d/13nn8oqmTe3C8LqghuNIrnWNtUuu7Dsa4NtqhI2AbeiA/edit?usp=sharing) (Archi)

[Cas Banque Coq](https://docs.google.com/document/d/1G9XDNO8q_1TOvek2dAY7A0lceDFjJWM1n1m1TRzeupo/edit) (Data)

Envoyer une étude de cas, avec pleins de trous béants nécessitant soit des éclaircissements de la part des examinateurs, soit des hypothèses argumentées.

Préparer entre examinateurs pour savoir qui joue quel rôle sur l'étude de cas, et fixer quelques hypothèses de travail.

## Phases

1. tour de table
2. Présentation de l'étude de cas préparée : roleplay en mode on est les clients.
3. ---------- Fin de la section préparée par le candidat ----------
4. Parler d'une expérience marquante : réalisation qu'il aurait mieux faite aujourd'hui, ou une expérience qu'on veut creuser.

Posture, clarté du discours, capacité de synthése (en live et en préparation), à faire dans les slides, à rebondir, réagir, argumenter, remise en question. Capacité à gérer son temps, surtout la présentation.

=> proposer du **schéma en live / outil de collaboration à distance** (proposer au candidat de travailler sur son outil et proposer un outil par défaut). Idées d’outils à dispo :

* <https://jamboard.google.com/> (intégré à la suite Google, on peut le créer un jam à la volée dans meet -> pas testé avec des adresses hors Octo)
* google slide
* <https://excalidraw.com/> (nickel pour des boiboites et des fleches)
* <https://awwapp.com/>