

Formation Kong Gateway – Développement

Réf. SF34574

Maîtrisez le workflow complet du développeur API avec Kong Gateway, de la conception avec Insomnia jusqu'au déploiement automatisé via APIOps.

2 j Durée	1 700 € HT Inter / participant	4 200 € HT Intra (≤12 pers.)	0 Sessions à venir
--------------	-----------------------------------	---------------------------------	-----------------------

Objectifs pédagogiques

- ✓ Concevoir et tester des APIs en suivant une approche API-first avec Insomnia et la spécification OpenAPI
- ✓ Gérer la configuration Kong Gateway de manière déclarative avec deck et les extensions x-kong
- ✓ Automatiser le déploiement des APIs en mettant en place un pipeline APIOps (CI/CD)
- ✓ Configurer les plugins de sécurité (API Key, JWT, mTLS, OpenID Connect) pour protéger ses APIs
- ✓ Implémenter le contrôle du trafic et la transformation des requêtes/réponses via les plugins Kong
- ✓ Mettre en place l'observabilité des APIs avec Prometheus, Grafana et la journalisation centralisée

Public visé & prérequis

Public visé Développeurs, Ingénieurs DevOps

Prérequis- Avoir suivi la formation "Kong Gateway : Fondations" (ou connaissances équivalentes : terminologie Kong, services, routes, plugins, consumers) - Connaissances de base en API REST et HTTP - Familiarité avec le format YAML - Familiarité avec Git et la gestion de version - Notions de spécification OpenAPI / Swagger

Programme détaillé

Conception API-first avec Insomnia et OpenAPI

- Approche API-first : concevoir le contrat d'interface avant d'écrire le code pour garantir la cohérence entre équipes
- Spécification OpenAPI : structure d'un fichier OpenAPI, versioning et bonnes pratiques de modélisation
- Workflow Insomnia : utiliser les modes Design, Debug et Test pour couvrir l'ensemble du cycle de conception
- Modélisation d'endpoints : définir les ressources, méthodes HTTP, schémas de requête et de réponse
- Tests automatisés avec Inso CLI : valider la conformité de l'API via des tests exécutables en ligne de commande
- Écosystème d'outils Kong pour le développeur : positionnement d'Insomnia, Inso, deck et leurs interactions dans le workflow quotidien

Activités pratiques :

- Conception d'une API REST complète dans Insomnia à partir d'une spécification OpenAPI
- Test et débogage d'endpoints avec les environnements Insomnia
- Exécution de suites de tests automatisés avec Inso CLI

Configuration déclarative avec deck

- Configuration déclarative vs impérative : comprendre l'intérêt du mode déclaratif pour la reproductibilité et la traçabilité
- Manifestes YAML deck : structure, syntaxe et organisation des fichiers de configuration
- Commandes essentielles deck : maîtriser dump, sync, diff et validate pour gérer le cycle de vie de la configuration
- Export et synchronisation : capturer l'état courant de Kong et le synchroniser depuis un dépôt de code
- Gestion multi-environnements : stratégies pour maintenir des configurations distinctes (dev, test, prod) avec deck

Activités pratiques :

- Export de la configuration courante avec `deck dump` et analyse du manifeste généré
- Création d'un manifeste YAML décrivant services, routes et plugins, puis synchronisation avec `deck sync`
- Comparaison d'états avec `deck diff` pour anticiper les changements avant déploiement

Extensions x-kong et spécification OpenAPI

- Directives x-kong : enrichir une spécification OpenAPI avec les annotations propriétaires Kong (x-kong-plugin-*, x-kong-route-defaults)
- Configuration intégrée à la spec : déclarer services, routes et plugins directement dans le fichier OpenAPI
- Approche fédérée : permettre à chaque équipe de gérer sa spécification et sa configuration Kong de manière autonome
- Lien spec-manifeste : comprendre la relation entre une spécification OpenAPI annotée et le manifeste deckK généré

Activités pratiques :

- Annotation d'une spécification OpenAPI existante avec des directives x-kong
- Génération automatique d'une configuration deckK à partir d'une spec annotée
- Configuration d'un plugin de sécurité directement dans la spécification OpenAPI

Automatisation des déploiements avec APIOps

- Concept APIOps : appliquer les principes GitOps au cycle de vie des APIs pour industrialiser les déploiements
- Pipeline CI/CD pour APIs : intégrer deckK et Inso dans GitHub Actions ou GitLab CI pour automatiser validation et déploiement
- Flux automatisé lint-test-diff-sync : chaîner les étapes de vérification, test, comparaison et synchronisation dans un pipeline
- Cycle de vie complet : orchestrer la transition de la conception au déploiement en production sans intervention manuelle

Activités pratiques :

- Mise en place d'un pipeline CI/CD intégrant la validation Inso et le déploiement deckK
- Automatisation du flux complet : commit d'une spec OpenAPI annotée déclenchant lint, tests et synchronisation Kong

Sécurisation des APIs côté développeur

- Authentification vs autorisation : distinguer les mécanismes d'identification et de contrôle d'accès dans Kong
- Plugin Key Auth : protéger une API par clé d'API, configuration et gestion des consumers
- Plugin JWT : valider des tokens JSON Web Token pour une authentification stateless
- Plugin Mutual TLS : mettre en place l'authentification mutuelle par certificat pour les échanges machine-to-machine
- Plugin OpenID Connect : déléguer l'authentification à un fournisseur d'identité externe (Keycloak, Auth0, etc.)
- Configuration déclarative de la sécurité : déclarer les plugins d'authentification via deckK pour une gestion versionnée

Activités pratiques :

- Configuration de chaque mode d'authentification (API Key, JWT, mTLS) via un manifeste deckK
- Test des accès sécurisés avec Insomnia et vérification du comportement en cas de credentials invalides

Contrôle du trafic et transformation des requêtes

- Rate Limiting : protéger les APIs contre les abus en limitant le débit par consumer, route ou service
- Proxy Caching : optimiser les performances en mettant en cache les réponses fréquemment sollicitées
- ACL (Access Control Lists) : restreindre l'accès à certaines routes en combinant groupes de consumers et listes blanches/noires
- Déploiements canari : router progressivement le trafic vers une nouvelle version d'API pour réduire les risques de mise en production
- Request Transformer : modifier les requêtes en transit (ajout, suppression ou réécriture de headers, body, query params)
- Response Transformer : adapter les réponses avant qu'elles n'atteignent le client (ajout de headers, restructuration du body)
- Manipulation JSON avec jq : transformer des payloads complexes grâce aux expressions jq intégrées aux plugins Kong
- Exit Transformer : personnaliser les réponses d'erreur pour offrir des messages cohérents et exploitables aux consommateurs d'API

Activités pratiques :

- Configuration d'un rate limiting et d'un proxy caching sur une route via deckK
- Mise en place d'un déploiement canari avec routage progressif du trafic
- Transformation d'une requête entrante (ajout de headers, réécriture de body) et personnalisation d'une réponse d'erreur avec Exit Transformer

Observabilité et monitoring des APIs

- Correlation ID : tracer une requête de bout en bout dans un environnement distribué grâce à un identifiant de corrélation
- Prometheus et Grafana : collecter les métriques Kong et les visualiser dans des dashboards temps réel
- Journalisation locale avec File Log : enregistrer les requêtes et réponses dans des fichiers pour le débogage
- Journalisation centralisée avec TCP Log et Splunk : agréger les logs dans une plateforme centralisée pour l'analyse à grande échelle
- Stratégie d'observabilité : combiner métriques, logs et traces pour couvrir les trois piliers de l'observabilité

Activités pratiques :

- Activation du Correlation ID et traçage d'une requête à travers plusieurs services
- Connexion de Prometheus à Kong et création d'un dashboard Grafana de supervision des APIs
- Configuration de la journalisation vers un fichier local et vers Splunk

Pourquoi choisir Sparks Formation ?

20 ans d'expertise IT, des formateurs experts et des sessions garanties contre l'annulation.

20 ans
d'expertise

1 068+
formations

11 178
apprenants formés/an

+67
Net Promoter Score

Avis des participants

Note moyenne des participants de la filière Développement et Ingénierie de Logiciels : 4,9/5 — 165 avis

Donatien W. — Formation dense mais très digeste avec beaucoup de points abordés. Les spécificités de chacun ont pu être abordées.

Jean-Baptiste M. — Le format 3 jours est une contrainte forte. A la sortie de cette formation, on attaque à peine la "vallée du doute" et on sent qu'il y a...

Votre formateur

Yann C. — Formateur Java

Yann intervient sur des formations portant sur l'architecture et le design logiciel. Les sujets abordés incluent notamment Java. Son activité de formation remonte à 2005, soit plus de 20 ans...

Ils nous font confiance

Cette année notre filière Architecture & Design Logiciel a accompagné plus de 50 entreprises dont :

SOPRA STERIA CIMPA SAS THALES GLOBAL SERVICES SAS ADEO SERVICES AUBAY LBC FRANCE ORANGE BUSINESS SQLI WORLDLINE FRANCE SAS ADEVINTA +40 autres

Financement & certifications

Cette formation est éligible aux financements OPCO et FNE-Formation. Nos équipes vous accompagnent dans le montage de votre dossier.



Réservez votre place dès maintenant

20 ans d'expertise · 1 000+ formations · NPS +67 · réponse sous 24h ouvrées

0 805 950 800

demande@sparks-formation.com