

Formation : REST API, conception, architecture et sécurité

Jour 1

- ORSYS
- Alex Zérah

Présentation

- Consultant en innovation technologique
- Développeur fullstack
- Formateur technologie web

Tour de table

Présentez-vous en incluant :

- La raison pour laquelle vous suivez cette formation
- Votre expérience en développement web
- Vos attentes pour cette formation

Planning

- 3 jours
- 6 séances
- 8 modules
- Théorie / Pratique

Plan de cours

- Introduction aux APIs ReST
- Bonnes pratiques
- La boîte à outils
- Rappels sur la sécurité
- Authentification et autorisation
- Middleware et JWT (JSON Web Token)
- Les Tests d'API
- API Management

Setup

Atelier : Mise en place du setup

Introduction aux APIs ReST

Architecture n-tiers, applications et APIs 1

Qu'est-ce qu'une API ?

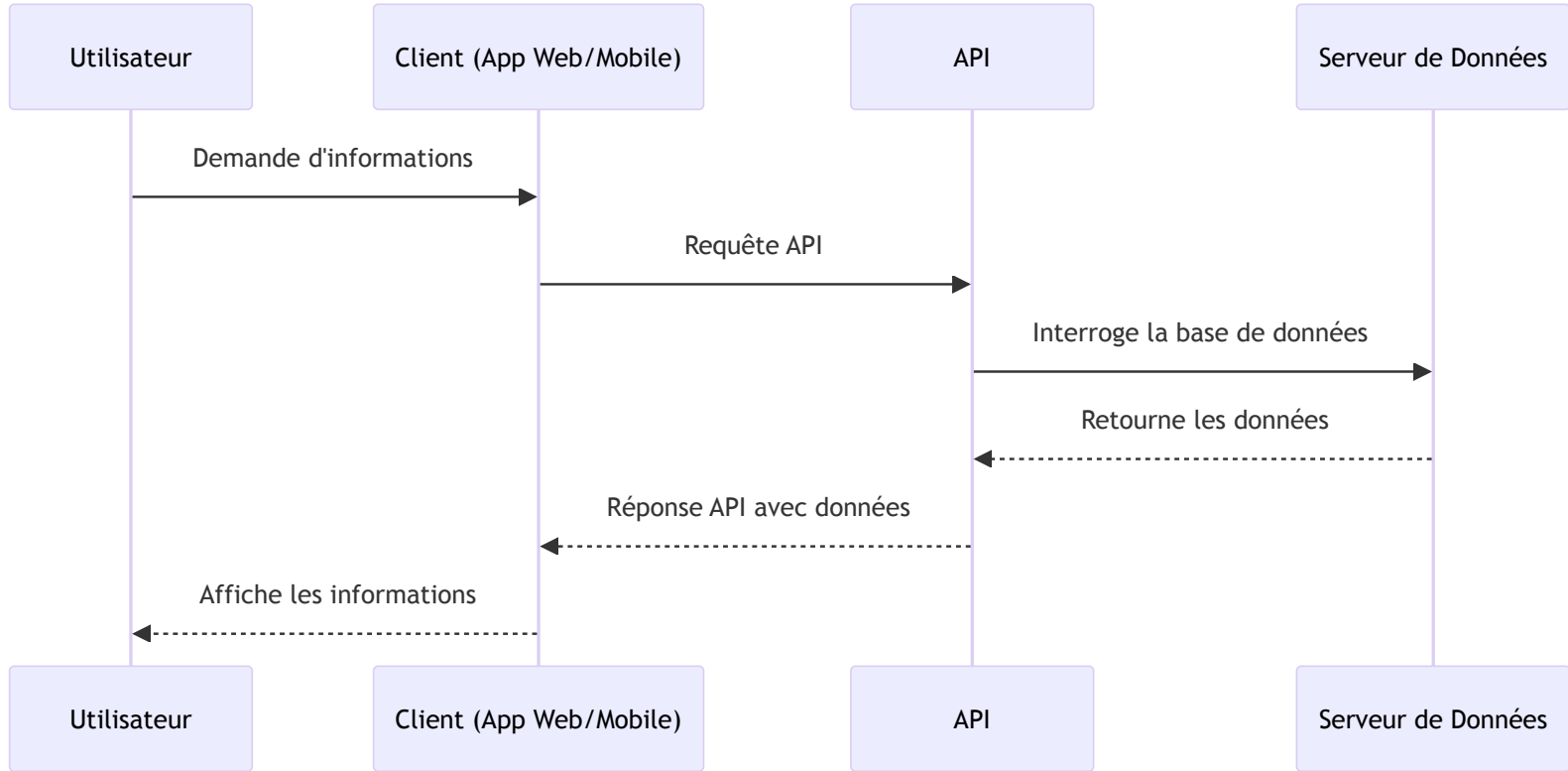
API = Application Programming Interface

Architecture n-tiers, applications et APIs 2

Une API est un ensemble de définitions et de protocoles qui facilite la création et l'intégration de logiciels d'applications.

Une API permet à un logiciel de communiquer avec un autre logiciel.

Architecture n-tiers, applications et APIs 3



Architecture n-tiers, applications et APIs 4

- **Utilisateur** : La personne utilisant l'application.
- **Client (App Web/Mobile)** : L'application que l'utilisateur utilise, qui envoie des requêtes à l'API.
- **API** : Interface qui reçoit les requêtes du client, les traite (en communiquant avec le serveur de données si nécessaire), et renvoie une réponse.
- **Serveur de Données** : Base de données ou autre source de données que l'API interroge pour récupérer les informations demandées.

Architecture n-tiers, applications et APIs 5

Par métonymie, on appelle souvent API l'ensemble de l'architecture, c'est-à-dire l'API elle-même, le serveur de données, et le client.

Architecture n-tiers, applications et APIs 6

Atelier : Concevoir une API pour la gestion d'un restaurant

Architecture n-tiers, applications et APIs 7

Architecture : Ensemble des composants d'un système informatique, leurs interrelations et les principes et lignes directrices qui gouvernent leur conception et leur évolution au fil du temps.

-> La structure de votre application

Architecture n-tiers, applications et APIs 8

Architecture 1-tier

- 1 serveur et client

Architecture n-tiers, applications et APIs 9

Architecture 2-tiers

- 1 client (présentation)
- 1 serveur (données et logique métier).

Architecture n-tiers, applications et APIs 10

Architecture 3-tiers

- 1 client (présentation)
- 1 serveur (logique métier).
- 1 serveur (accès aux données)

Architecture n-tiers, applications et APIs 11

Architecture n-tiers

- 1 client (présentation)
- 1 serveur (logique métier).
- 1 serveur (accès aux données)
- Composants supplémentaires

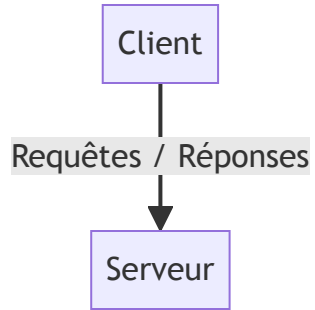
Architecture n-tiers, applications et APIs 12

1-tier

Client / Serveur / Base de Données

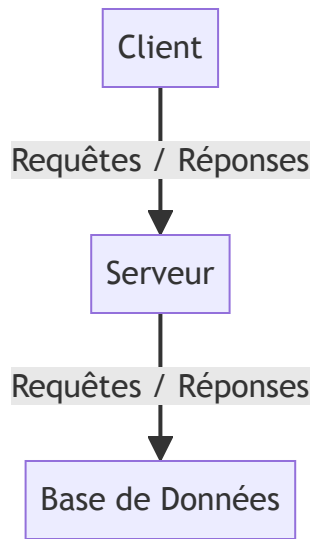
Architecture n-tiers, applications et APIs 13

2 tiers



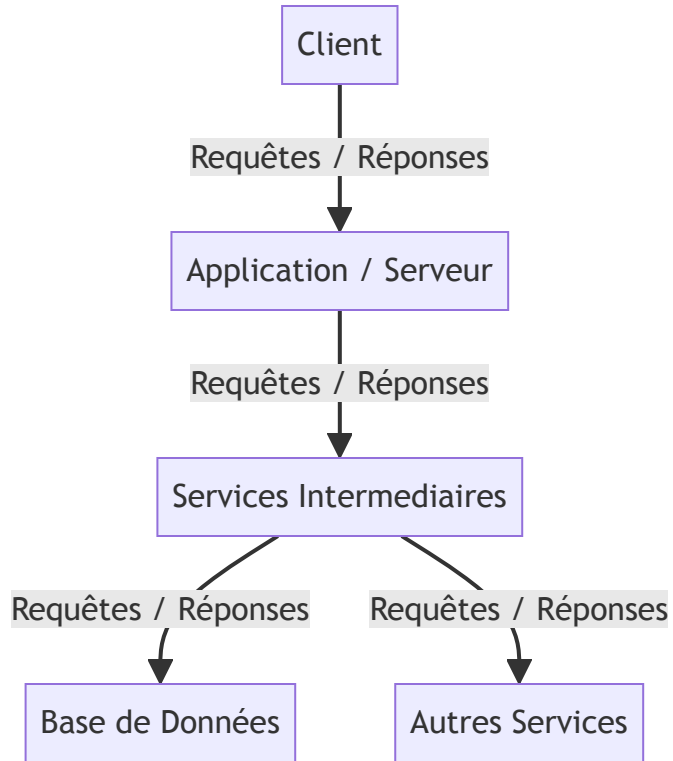
Architecture n-tiers, applications et APIs 14

3 tiers

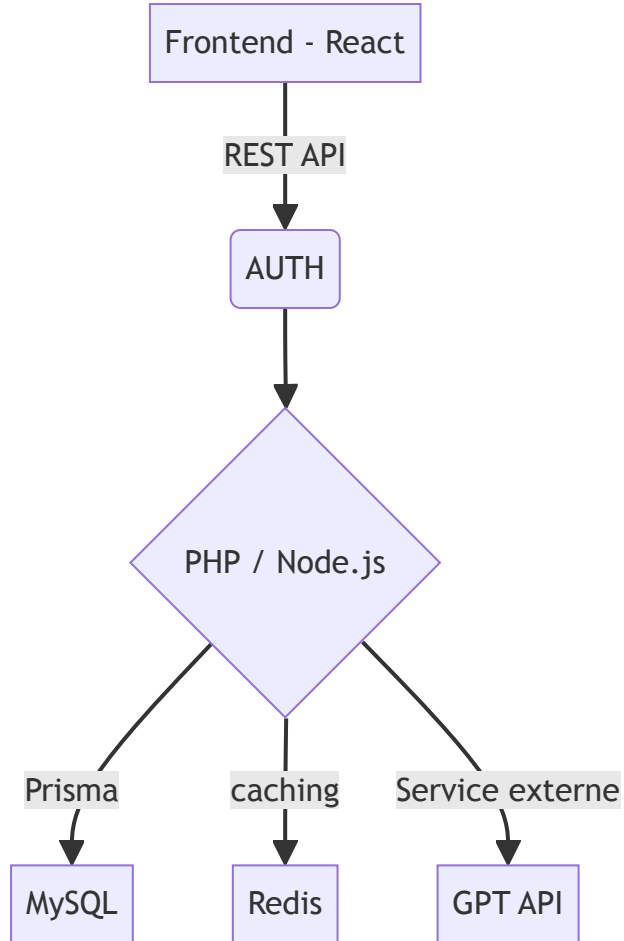


Architecture n-tiers, applications et APIs 15

n tiers



Architecture n-tiers, applications et APIs 16



Architecture n-tiers, applications et APIs 17

Résumé

Les différences essentielles entre une API REST et une API SOA 1

REST et SOA : styles d'architecture logicielle

Les différences essentielles entre une API REST et une API SOA

2

API REST

- REST : Representational State Transfer
- Créé par Roy Fielding en 2000
- Défini un ensemble de contraintes et de propriétés basées sur le protocole HTTP
- JSON

Les différences essentielles entre une API REST et une API SOA

3

API SOA

- SOA : Service Oriented Architecture
- Créé par IBM en 1996
- Défini un ensemble de principes et de bonnes pratiques
- Enveloppe et modèle de données
- XML

Les différences essentielles entre une API REST et une API SOA

4

Source : AWS - difference between SOAP and REST

Les différences essentielles entre une API REST et une API SOA

5

Atelier : Créer une API REST

H.A.T.E.O.A.S. Gestion des ressources et liens hypermedia 1

- HATEOAS (Hypermedia as the Engine of Application State)
- Contrainte de l'architecture REST
- Ressources et liens hypermedia

H.A.T.E.O.A.S. Gestion des ressources et liens hypermedia 2

```
{
  "platId": "123",
  "nom": "Poulet Rôti",
  "description": "Poulet rôti avec herbes et épices, servi avec des légumes de saison",
  "prix": 15.99,
  "_links": {
    "self": {
      "href": "http://api.monrestaurant.com/plats/123"
    },
    "commander": {
      "href": "http://api.monrestaurant.com/plats/123/commander"
    },
    "platsSimilaires": {
      "href": "http://api.monrestaurant.com/plats/similaires?platId=123"
    },
    "tousLesPlats": {
      "href": "http://api.monrestaurant.com/plats"
    }
  }
}
```

H.A.T.E.O.A.S. Gestion des ressources et liens hypermedia 3

- ``self`` : Lien vers la ressource elle-même
- ``commander`` : Lien pour commander ce plat
- ``platsSimilaires`` : Lien vers une liste de plats similaires
- ``tousLesPlats`` : Lien vers la liste de tous les plats
- ``href`` : URL du lien

H.A.T.E.O.A.S. Gestion des ressources et liens hypermedia 4

Atelier : Ajouter des liens hypermedia à votre API

2. Bonnes pratiques

2.1 Conventions, bonnes pratiques et ReSTafarians 1

- Convention et bonnes pratiques : créer une API REST robuste, maintenable et évolutive

2.1 Conventions, bonnes pratiques et ReSTafarians 2

- Objectif REST : développer l'API de manière pragmatique.
- Ce n'est pas un dogme
- Ne pas être "ReSTafarians"

2.1 Conventions, bonnes pratiques et ReSTafarians 3

- Liste des conventions et bonnes pratiques

2.1 Conventions, bonnes pratiques et ReSTafarians 4

Atelier : Mettre en place des conventions et bonnes pratiques dans votre API

2.2 Techniques et Stratégies de Versioning 1

- Versioning : Gestion des versions de l'API
- L'API évolue au fil du temps
- Aider les clients à migrer vers la nouvelle version
- Maintenir la compatibilité avec les anciennes versions

2.3 Bonnes approches de conception et de développement 1

Bonnes pratiques : Outils et techniques pour développer une API REST dans les règles de l'art

Objectifs :

- compréhension, clarté
- maintenabilité
- évolutivité
- UX / DX

2.3 Bonnes approches de conception et de développement 2

- Liste bonnes approches de conception et de développement

2.3 Bonnes approches de conception et de développement 3

Atelier : Mettre en place des bonnes approches de conception et de développement dans votre API

3. La boîte à outils

3.1 API Mock 1

- API Mock : API factice qui simule le comportement d'une API réelle
- Utilisé pour tester une API

3.1 API Mock 2

Objectifs :

- Tester l'API avant qu'elle ne soit prête
- Programmer les réponses de l'API
- Développement du projet en parallèle (client et serveur)

3.1 API Mock 3

- Atelier : Créer une API Mock

3.2 Conception d'APIs ReST avec OpenAPI et Swagger 1

- OpenAPI : Spécification pour décrire des APIs ReST
- Swagger : Outil pour concevoir des APIs ReST

3.2 Conception d'APIs ReST avec OpenAPI et Swagger 2

Atelier : Ajouter à l'API avec OpenAPI et Swagger

3.3 Utilisation de Postman ou Insomnia 1

- Postman : Outil pour tester des APIs ReST

3.3 Utilisation de Postman ou Insomnia 2

- Permet de faire des requêtes HTTP (GET, POST, PUT, DELETE, PATCH, OPTIONS, HEAD)
- Permet de tester les APIs de manière visuelle
- Utile au développement

3.3 Utilisation de Postman ou Insomnia 3

Atelier : Tester l'API avec Postman

3.4 Environnement de test et outils (JSON Generator. JSON Server) 1

- JSON Generator : Générateur de données aléatoires
- JSON Server : Serveur de données en JSON

3.4 Environnement de test et outils (JSON Generator. JSON Server) 2

Atelier : Créer un environnement de test avec JSON Generator et JSON Server

Conclusion journée