

**SQLI
DIGITAL
EXPERIENCE**



API DESIGN FIRST

**BOOSTEZ LE DÉVELOPPEMENT
DE VOS APIS**

DÉMARCHE API

Normalisation

- Définition des standards du SI et normalisation

Architecture

- Approche Domain Driven Design et Architecture Hexagonale

Spécification

- Approche Design First pour les spécifications et la documentation

Implémentation

- Se baser sur de bonnes pratiques de développement

Testabilité

- Garantir la qualité au travers de la pyramide de tests

Management

- Mise en place d'outils de gouvernance des API

Observabilité

- Mise en place d'outils d'observabilité et de monitoring

CODE FIRST

- ON CODE DANS SON IDE
- ON TESTE VIA POSTMAN
- LA DOCUMENTATION ?
- COMMUNIQUER AUX CONSOMMATEURS ?

```
@RestController
@RequestMapping("/{api/v1/hello"})
public class HelloAPI {

    private static final Logger LOGGER =
        LoggerFactory.getLogger(HelloAPI.class);

    @GetMapping
    public ResponseEntity<HelloDto> hello() {
        LOGGER.info("GET /api/v1/hello");
        HelloDto result = new HelloDto();
        result.setMessage("Hello World");
        LOGGER.info("Response : " + result.getMessage());
        return ResponseEntity.ok(result);
    }
}
```

LE DESIGN FIRST

- IMPLÉMENTER PLUS EFFICACEMENT VOS APIS
- INTÉGRER PLUS FACILEMENT LES APPELS À VOS APIS
- POSSIBILITÉ DE SIMULER VOS APIS
- DOCUMENTATION EN ADÉQUATION AVEC L'IMPLÉMENTATION RÉELLE

OPENAPI

Version	Date	Notes
1.0	2011-08-10	First release of the Swagger Specification
2.0	2014-09-08	Release of Swagger 2.0
2.0	2015-12-31	Donation of Swagger 2.0 to the OpenAPI Initiative
3.0.0	2017-07-26	Release of the OpenAPI Specification 3.0.0
3.1.0	2021-02-15	Release of the OpenAPI Specification 3.1.0

<http://spec.openapis.org/oas/v3.1.0>

DÉFINIR SON API



- LES ROUTES
- LES VERBES HTTP
- LES PARAMÈTRES ET RÉPONSES
- LES CONTRÔLES
- LA SÉCURITÉ
- LES CODES RETOURS
- LA DOCUMENTATION

The screenshot displays the Swagger Editor interface. On the left, the OpenAPI 3.0.1 definition is shown in a code editor. The definition includes the following details:

- Info:**
 - title: Connaissance Client API
 - description: Documentation de l'API Connaissance Client
 - version: 1.0.0
- contact:**
 - name: SQLI
 - url: http://sqli.com/
 - email: pbousquet@sqli.com
- license:**
 - name: Copyright (c) SQLI
- servers:**
 - url: http://localhost/
 - description: DEV Server
 - variables: { }
 - url: https://jenesaispas/
 - description: Azure Cloud
 - variables: { }
- security:**
 - Authorization: []
- tags:**
 - name: connaissanceclient
 - description: opérations de lecture / écriture sur les fiches de connaissance client
- paths:**
 - /connaissanceClient:**
 - get:**
 - tags: connaissanceclient-api-controller
 - summary: Liste des fiches de connaissance client
 - operationId: getConnaissanceClients
 - description: liste des fiches de connaissance client
 - parameters:
 - in: query
 - name: sorted
 - description: sort the list of results
 - required: false
 - schema: type: boolean, default: false
 - responses:
 - 200: description: Réponse pour une requête valide, content: application/json, schema: \$ref: '#/components/schemas/ConnaissanceClients'
 - 400: description: Invalid Request, content: application/json, schema: \$ref: '#/components/schemas/ApiResponse'
 - 401: description: Access forbidden, content: application/json, schema: \$ref: '#/components/schemas/ApiResponse'

On the right, the rendered API documentation is shown. It features the title 'Connaissance Client API' with version '1.0.0' and 'OAS3' badge. Below the title, there are links for 'SQLI - Website' and 'Send email to SQLI', and a 'Copyright (c) SQLI' notice. A 'Servers' dropdown menu is set to 'http://localhost/ - DEV Server', with an 'Authorize' button. The main content area lists the API endpoints under the tag 'connaissanceclient'.

Method	Endpoint	Description	Lock
GET	/connaissanceClient	Liste des fiches de connaissance client	🔒
POST	/connaissanceClient	Accès en écriture à une fiche de connaissance client	🔒
POST	/connaissanceClientAS	Accès en écriture à une fiche de connaissance client	🔒
GET	/connaissanceClient/{id}	Accès en lecture à une fiche de connaissance client	🔒
PUT	/connaissanceClient/{id}/adresse	Changement d'adresse du client	🔒
PUT	/connaissanceClient/{id}/situation	Changement d'adresse du client	🔒

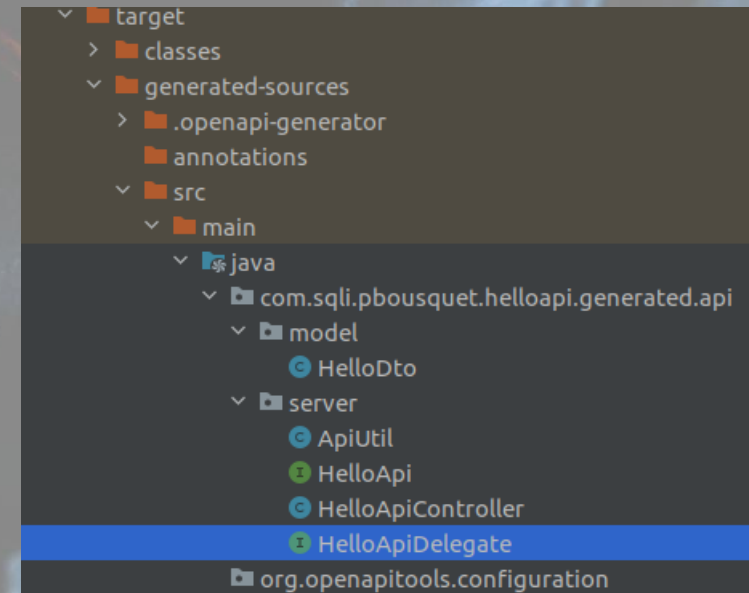
<https://editor.swagger.io/>

DÉMO



GÉNÉRATION SERVEUR

- L'EXPOSITION API (CONTRÔLEURS)
- LES DTOS
- LES CONTRÔLES
- LES TESTS UNITAIRES
- JAVA, SPRING, C#, PHP, NODE, ...



Open API Generator

IMPLÉMENTER

- LE DELEGATE
- CODE MÉTIER
- COUPLAGE FAIBLE

```
@Component
public class HelloApiDelegateImpl implements HelloApiDelegate {

    @Override
    public ResponseEntity<HelloDto> helloWorld() {
        HelloDto helloDto = new HelloDto();
        helloDto.setMessage("Hello World");
        return ResponseEntity.ok(helloDto);
    }

    @Override
    public ResponseEntity<HelloDto> helloWithName(String name) {
        HelloDto helloDto = new HelloDto();
        helloDto.setMessage("Hello " + name);
        return ResponseEntity.ok(helloDto);
    }
}
```

<http://localhost:8080/swagger-ui.html>

DÉMO



GÉNÉRATION CLIENT

- LES DTOS
- LE CODE D'APPEL API
- JS, ANGULAR, JAVA, ...

```
▼ src
  ▼ app
    ▼ hello-api
      > .openapi-generator
      > api
      > model
      ◆ .gitignore
      ≡ .openapi-generator-ignore
      TS api.module.ts
      TS configuration.ts
      TS encoder.ts
      $ git_push.sh
      TS index.ts
      TS param.ts
      ⓘ README.md
      TS variables.ts
      TS app-routing.module.ts
      <> app.component.html
```

Open API Generator

INTÉGRER L'APPEL API



- IMPORTER LE SERVICE
- INSTANCIER LE SERVICE
- CODER L'APPEL

```
import { Component } from '@angular/core';
import { HelloService } from '../hello-api/api/hello.service';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss'],
})
export class AppComponent {
  title = 'Appel de l\'api hello';
  result = this.helloService.helloUsingGET1().subscribe(
    helloDto => (this.title = helloDto.message!)
  );
  constructor(private helloService: HelloService){}
}
```

<http://localhost:4200/>

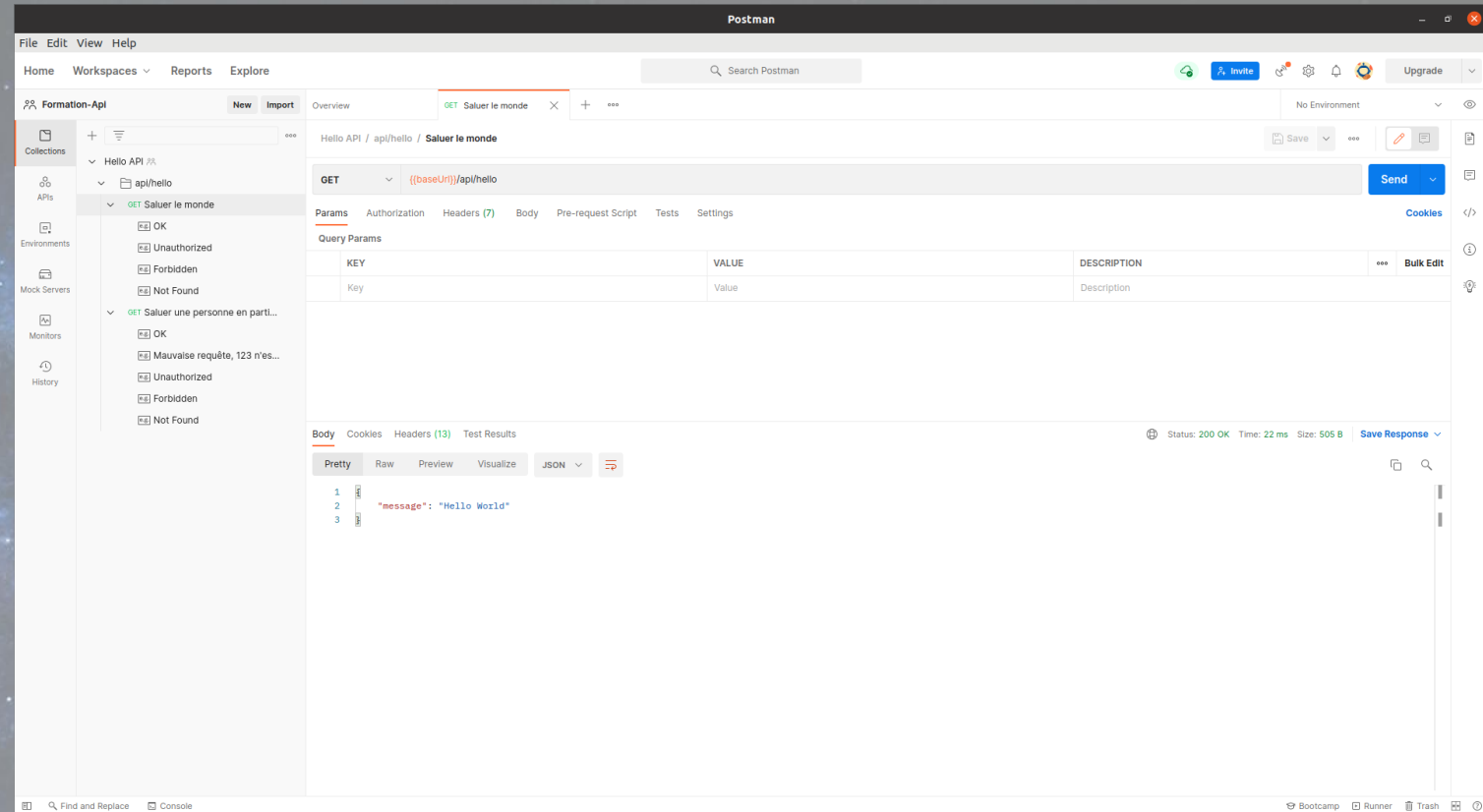
DÉMO



TESTS & MOCKS



- COLLECTIONS DE TESTS
- MOCK SERVER
- AUTRES...



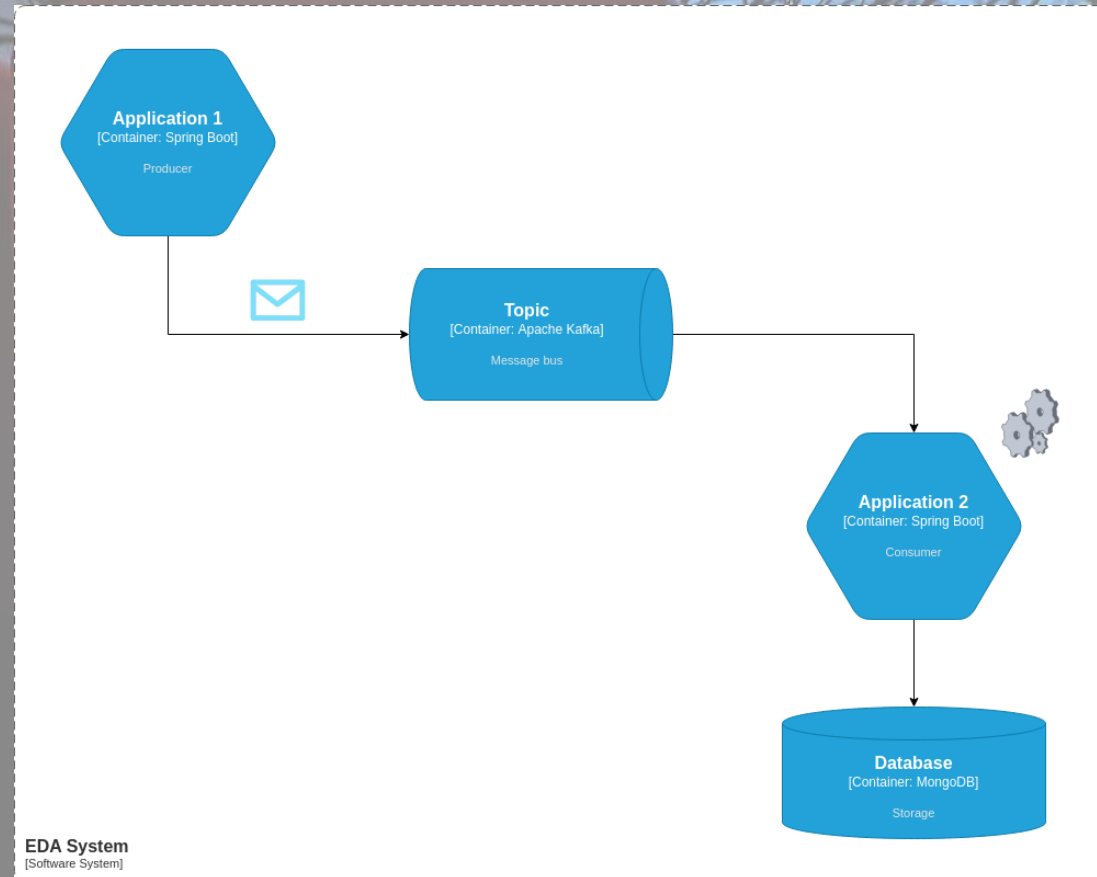
<https://openapi.tools/>

DÉMO



L'ASYNCHRONE

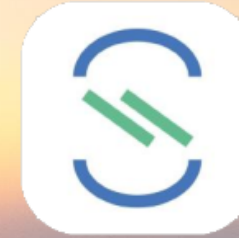
- SPÉCIFICATIONS
- PRODUCER / CONSUMER
- MESSAGES
- GENERATION
- KAFKA, EVENT HUB, ...



<https://www.asyncapi.com/>

POUR CONCLURE

- OPENAPI ET ASYNCAPI
- IMPLÉMENTER PLUS EFFICACEMENT VOS APIS
- INTÉGRER PLUS FACILEMENT LES APPELS À VOS APIS
- POSSIBILITÉ DE SIMULER VOS APIS
- DOCUMENTATION EN ADÉQUATION AVEC L'IMPLÉMENTATION RÉELLE



MERCI



Philippe Bousquet
Architecte – Directeur Technique Java



SQLI DIGITAL EXPERIENCE

Retrouvez la présentation ici



➤ RENDEZ VOUS LE 10/11 SUR LE STAND SQLI À BDX I/O

BDX I/O