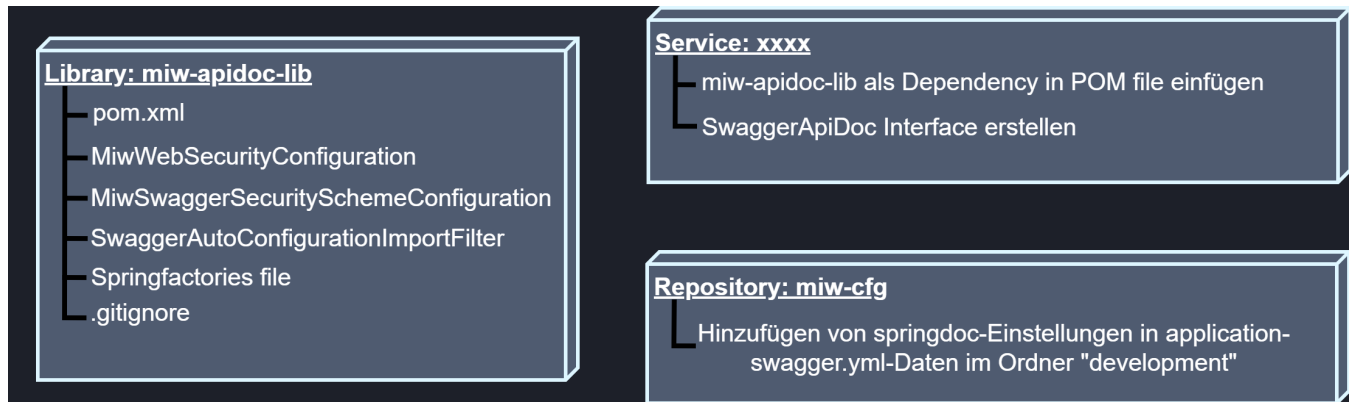


# Automatisierung der API-Spezifikation

Laut Enterprise Architekt Management (EAM) müssen wir OpenAPI Specification standard für die Automatisierung & die Visualisierung der API Spezifikation implementieren. Unten ist eine schematische visuelle Darstellung der Implementierung von OpenAPI in der Middleware zu sehen. Eine beschreibende Dokumentation findet Ihr unter der visuellen Darstellung.

Der Mastermind hinter dem Design und der Architektur gemäß OpenApi 3.0 im Middleware Team ist [Kavivarman Sivarasah](#)

Die Umsetzung dieses Design erfolgte durch [Pradeep Harikrishnan](#)



## Die folgenden vier Komponenten / Dateien wurden der Bibliothek „miw-apidoc-lib“ hinzugefügt:

- pom.xml: Diese Datei enthält die Dependencies, die zur Automatisierung und Visualisierung der API gemäß OpenAPI 3-Standard erforderlich sind.
- MiwWebSecurityConfiguration: Diese Klasse gewährt Zugriff auf SwaggerUI ohne Authentifizierung.
- MiwSwaggerSecuritySchemeConfiguration: Diese Klasse ermöglicht JWT Bearer Authentifikation.
- SwaggerAutoConfigurationImportFilter: Diese Klasse schließt alle Swagger Konfigurationen in Test-, Staging- und Produktionsumgebungen.
- Springfactories Datei:
- Gitignore: Alle Dateien, die für die von uns programmierte Software / Bibliothek nicht relevant sind, können in diese Datei verschoben werden.

## Konfiguration im miw-cfg Repository:

- Hinzufügen von springdoc-Einstellungen, um die Generierung von API-Dokumenten aus dem Dienst zu ermöglichen, die Visualisierung von API über swaggerUI und das Deaktivieren der Swagger-Standard-URL, die zum Petstore führt. Die Einstellungen müssen in die Entwicklungsumgebung in den Daten application-swagger.yml im development ordner integriert sein.

## Konfiguration im Service

- Hinzufügen der miw-apidoc-lib als Dependency und Konfiguration der entsprechenden Einstellungen im parent POM des Dienstes
- Trennung des technischen Codes vom Dokumentationscode durch Erstellung einer Schnittstelle (SwaggerApiDoc)

## Funktionsumfang:

lfd. Nr.	Funktion	Implementierung durch
1.	Automatisierung und Visualisierung der API gemäß Openapi 3 Standard	Verwendung der Dependencies in der POM-Datei
2.	Der Zugriff auf die SwaggerUI ohne Authentifizierung	Implementierung der Klasse MiwWebSecurityConfiguration
3.	Try it out function - einen Request direkt von SwaggerUI an den Service schicken	Implementierung der Klasse MiwSwaggerSecuritySchemeConfiguration
4.	Blockieren des Zugriffs / Deaktivierens von SwaggerUI & ApiDocs in verschiedenen Umgebungen	Implementierung der Klasse SwaggerAutoConfigurationImportFilter
5.	Deaktivierung der Swagger Standard URL zur Pet-Store	Konfiguration in application-swagger.yml Datei im development Ordner in miw-cfg Repo
6.	Kennzeichnung aller Pflichtfelder im Request Body	Implementierung einer Schnittstelle im entsprechenden Service (abhängig von der Architektur des Services)

7.	Beschreibung des Services und jeder Operation in der SwaggerUI	Implementierung einer Schnittstelle in der entsprechende Service durch Trennen des technischen Codes mit dem Dokumentationscode
8.	Definieren des Medientyps des Request Bodys und des Reponse Bodys	Die Implementierung dieser Funktion erfolgt im Service