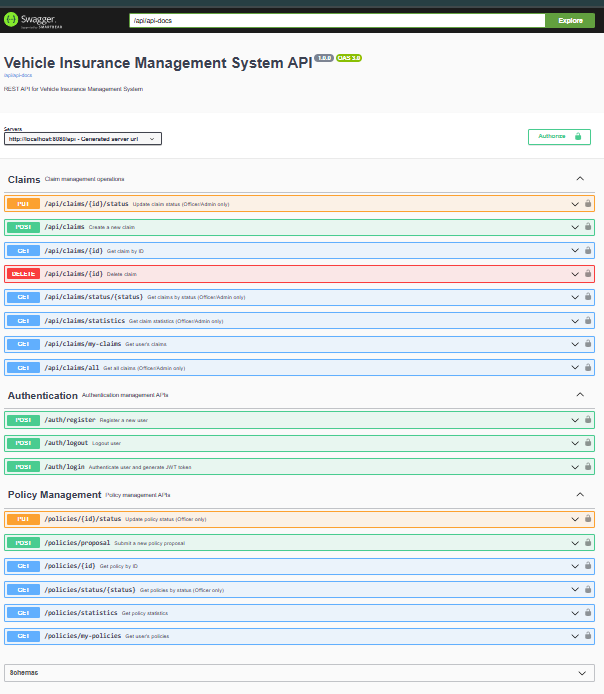
**Vehicle Insurance Management System API**

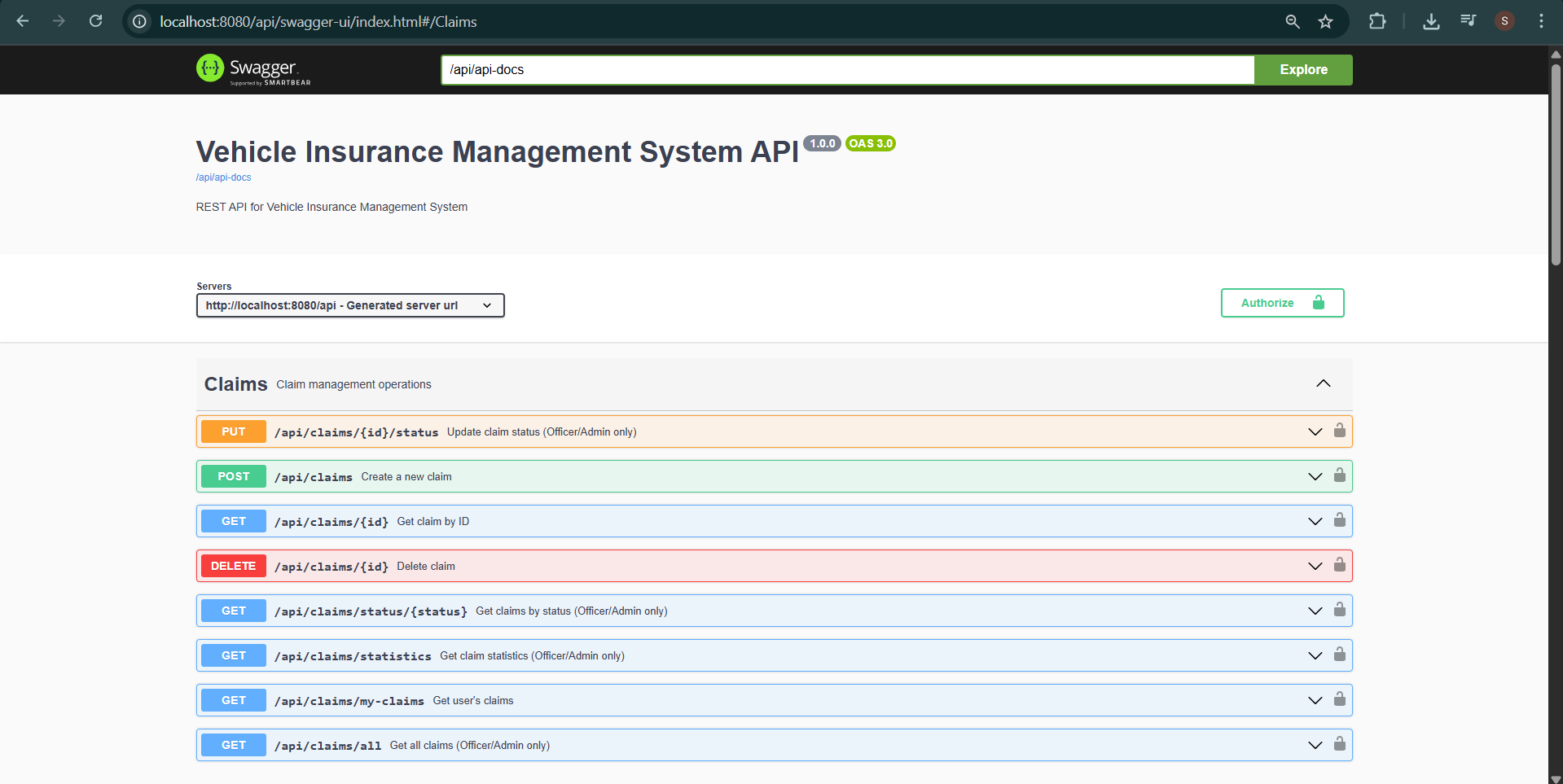
[**http://localhost:8080/api/swagger-ui/index.html#/Authentication/registerUser**](http://localhost:8080/api/swagger-ui/index.html#/Authentication/registerUser)

**swagger-ui**

****

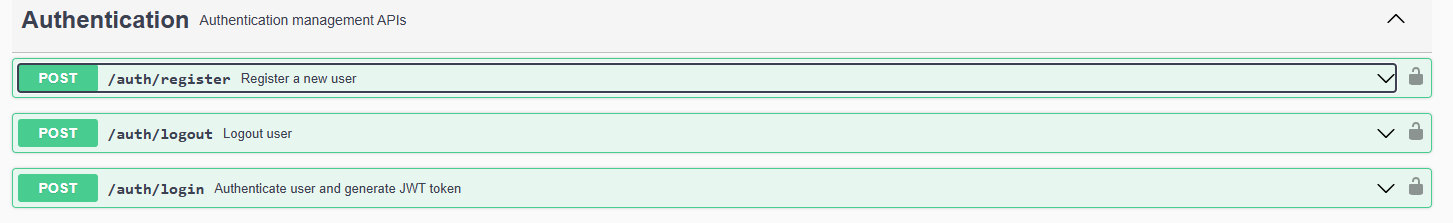
[**Claims**](http://localhost:8080/api/swagger-ui/index.html#/Claims)

Claim management operations



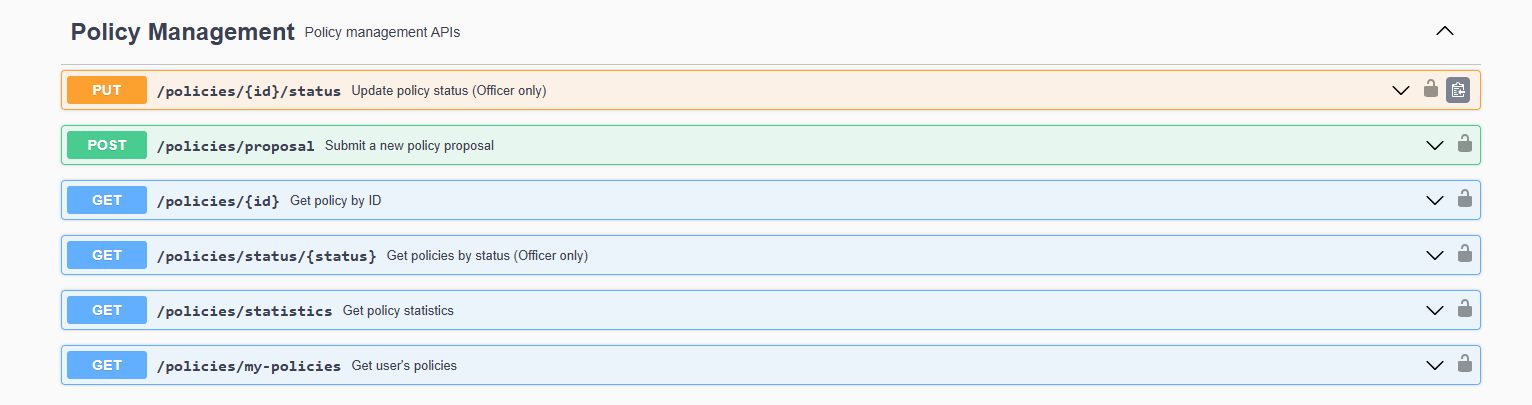
[**Authentication**](http://localhost:8080/api/swagger-ui/index.html#/Authentication)

Authentication management APIs



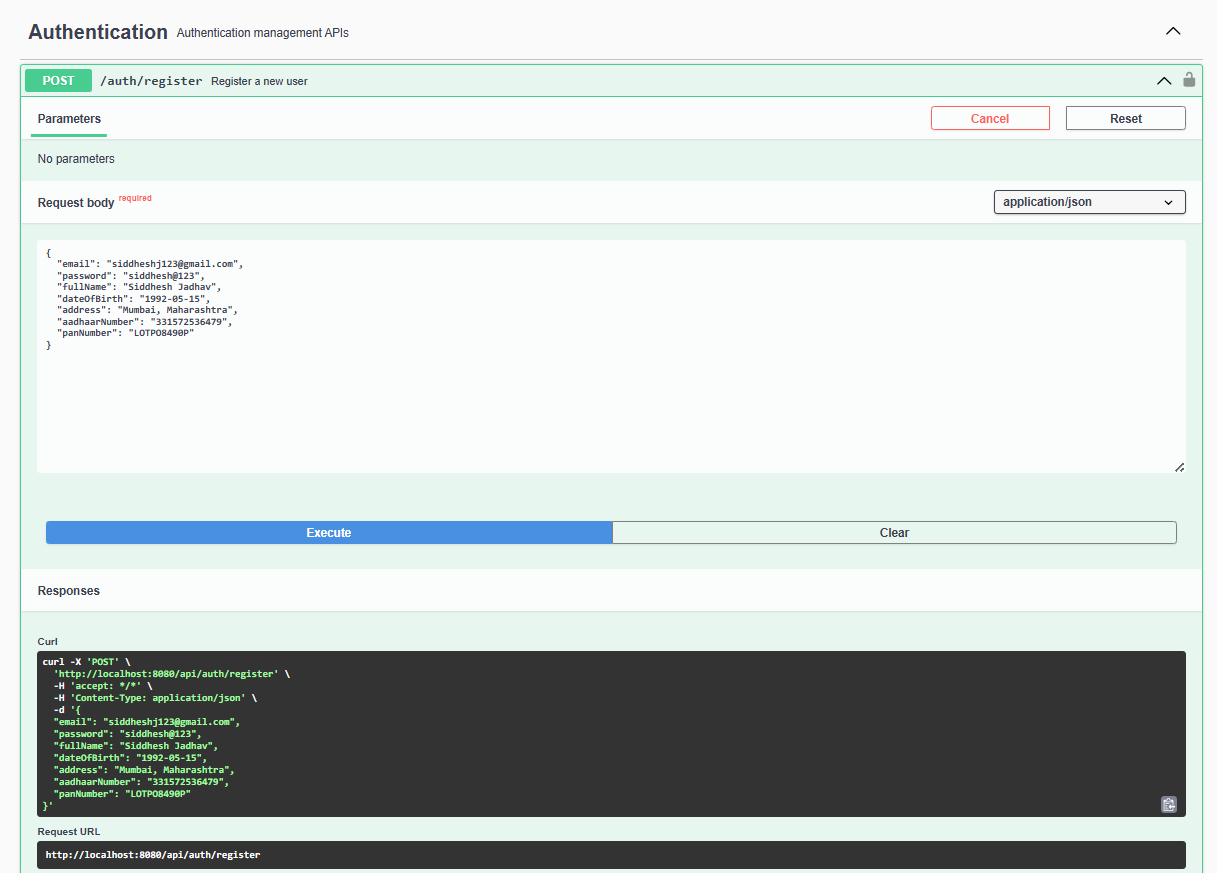
[**Policy Management**](http://localhost:8080/api/swagger-ui/index.html#/Policy%20Management)

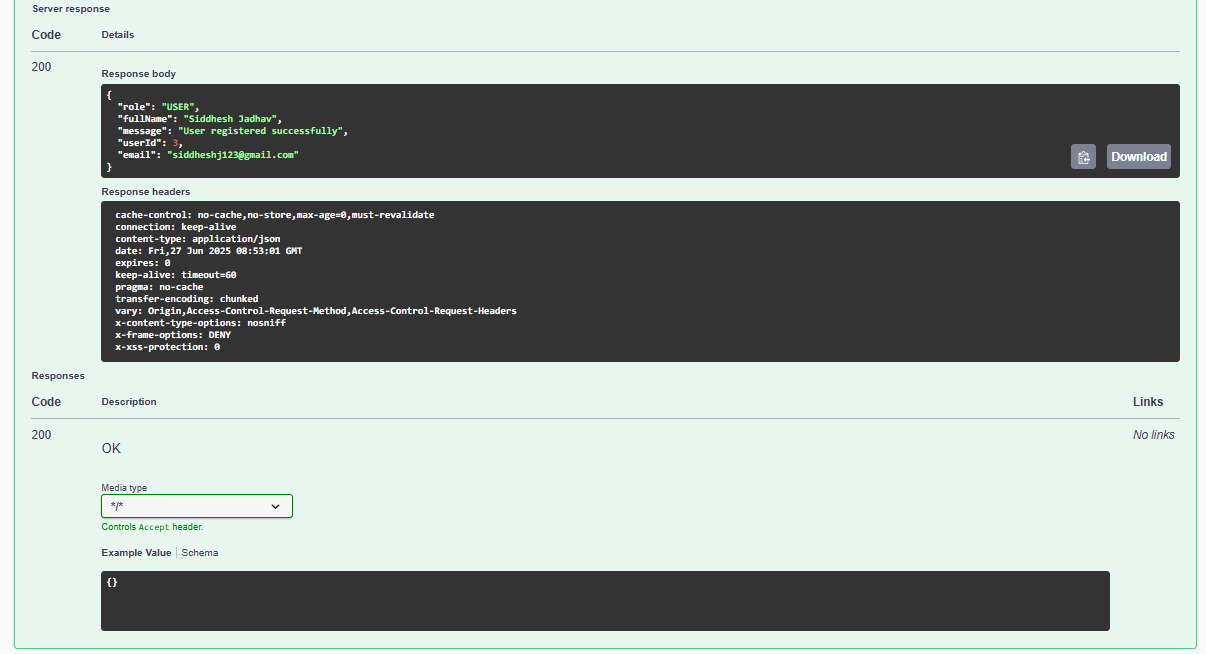
Policy management APIs



**Phase 1: Authentication & User Management**

**POST /api/auth/register**





{

"email": "siddheshj123@gmail.com",

"password": "siddhesh@123",

"fullName": "Siddhesh Jadhav",

"dateOfBirth": "1992-05-15",

"address": "Mumbai, Maharashtra",

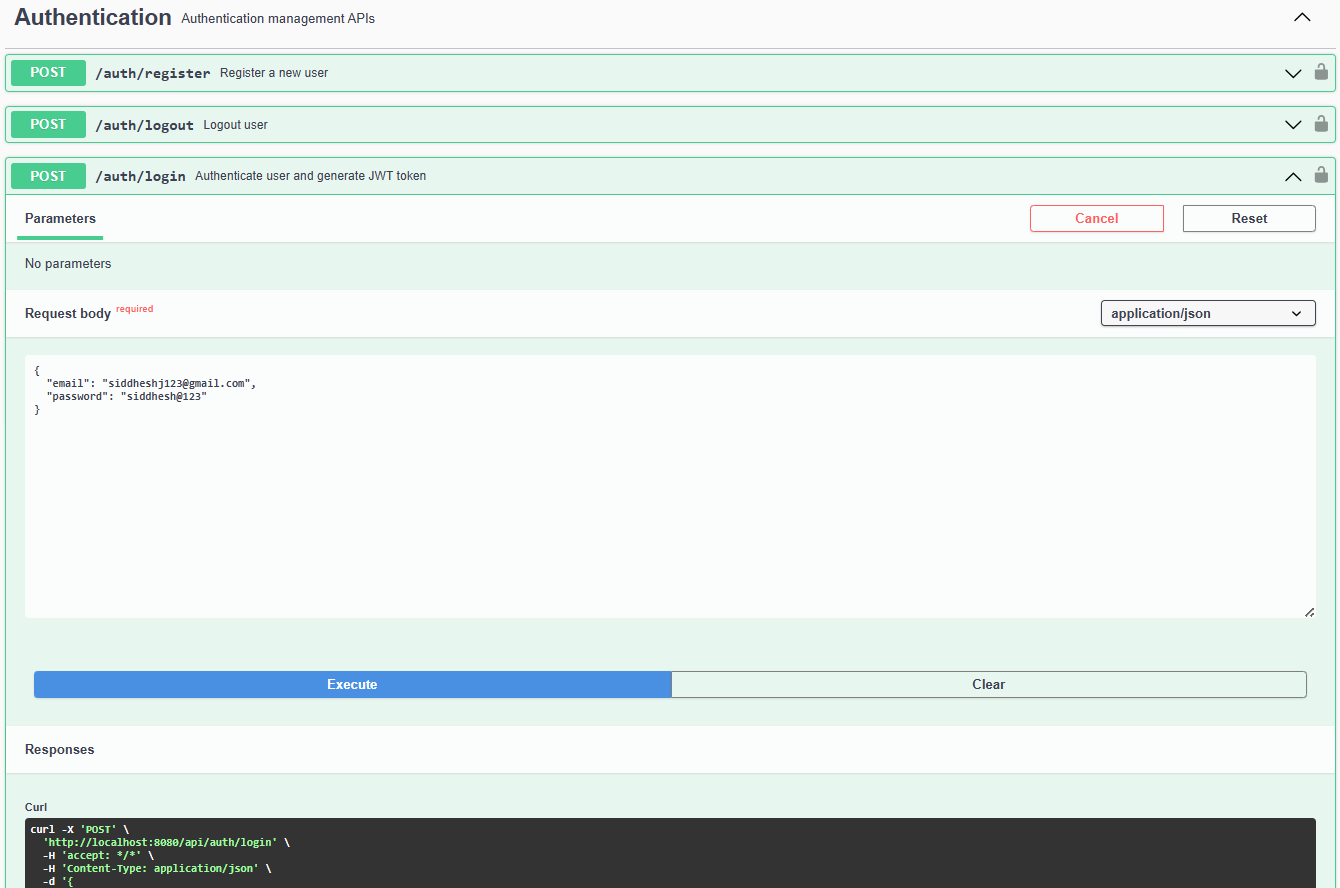
"aadhaarNumber": "331572536479",

"panNumber": "LOTPO8490P"

}

**1.2 Login User**

**POST /api/auth/login**

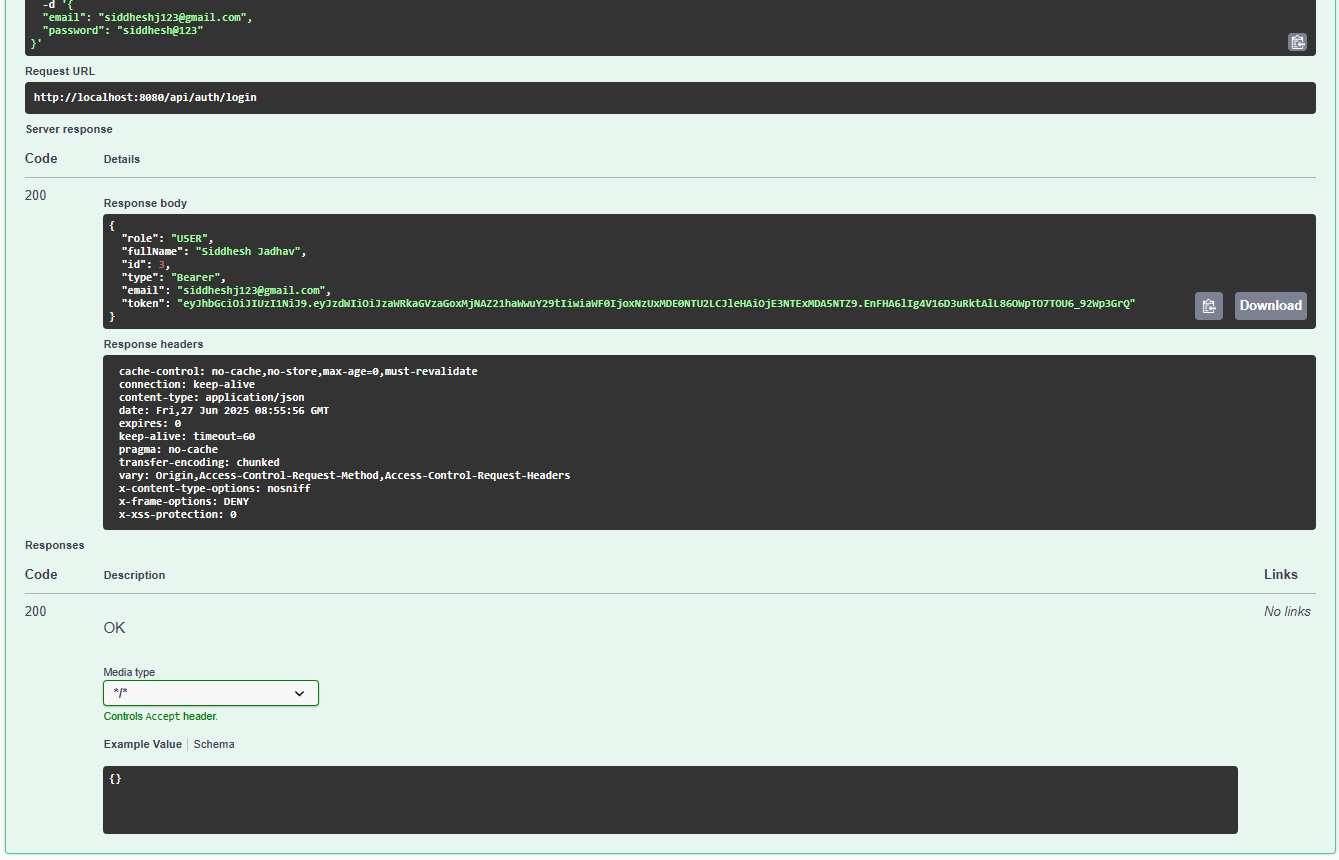


{

"email": "siddheshj123@gmail.com",

"password": "siddhesh@123"

}



Token generated

##### Response body

**{**

**"role": "USER",**

**"fullName": "Siddhesh Jadhav",**

**"id": 3,**

**"type": "Bearer",**

**"email": "siddheshj123@gmail.com",**

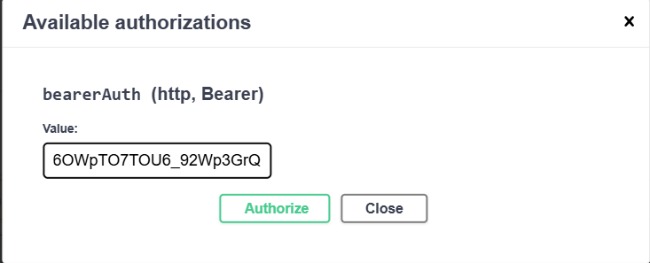
**"token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJzaWRkaGVzaGoxMjNAZ21haWwuY29tIiwiaWF0IjoxNzUxMDE0NTU2LCJleHAiOjE3NTExMDA5NTZ9.EnFHA6lIg4V16D3uRktAlL86OWpTO7TOU6\_92Wp3GrQ"**

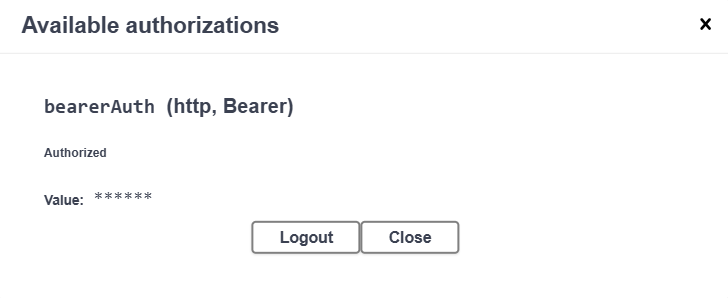
**}**

**Copy the JWT token from response!**

**1.3 Authorize (Required for all protected endpoints)**

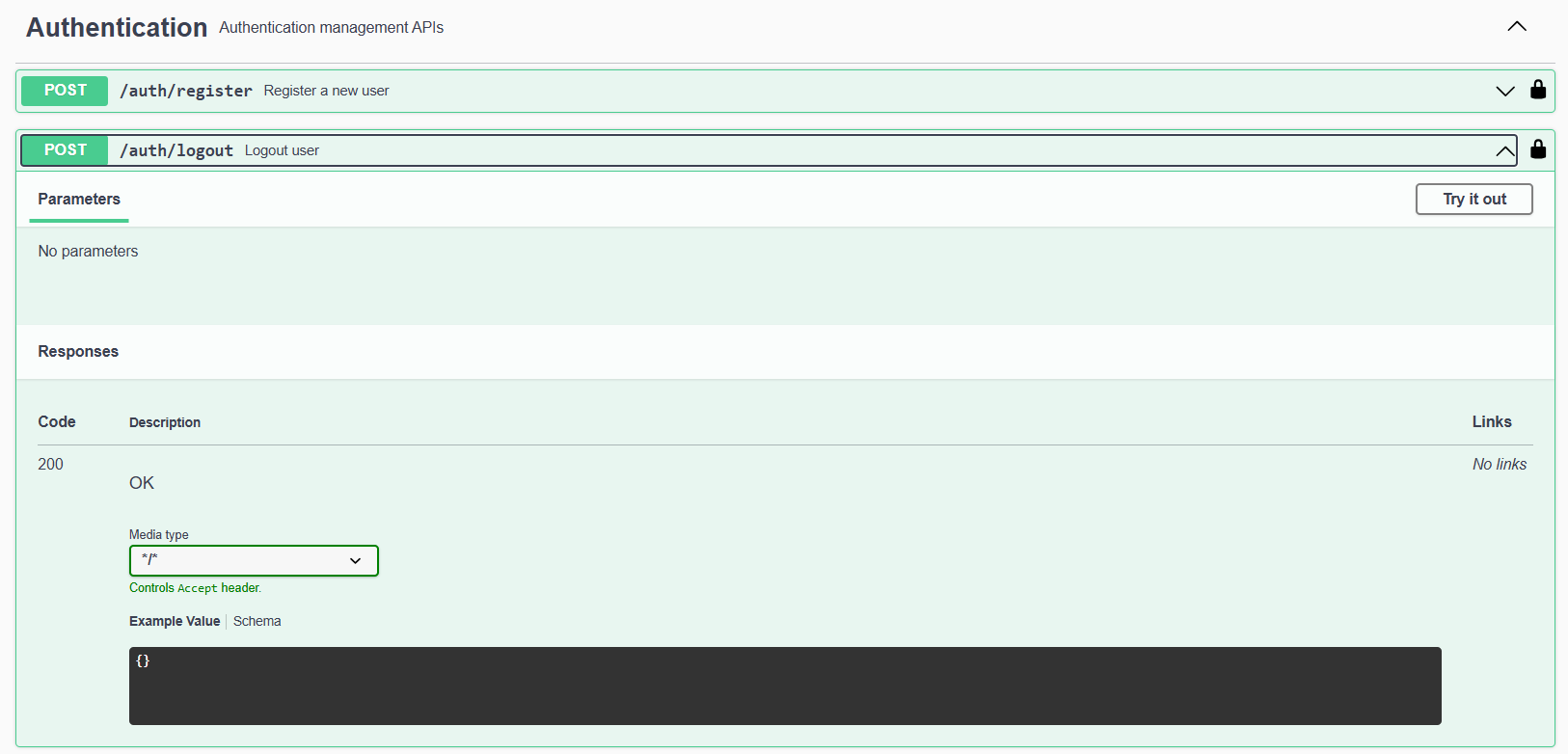
Click 🔒 **"Authorize"** → Enter: Bearer YOUR\_JWT\_TOKEN





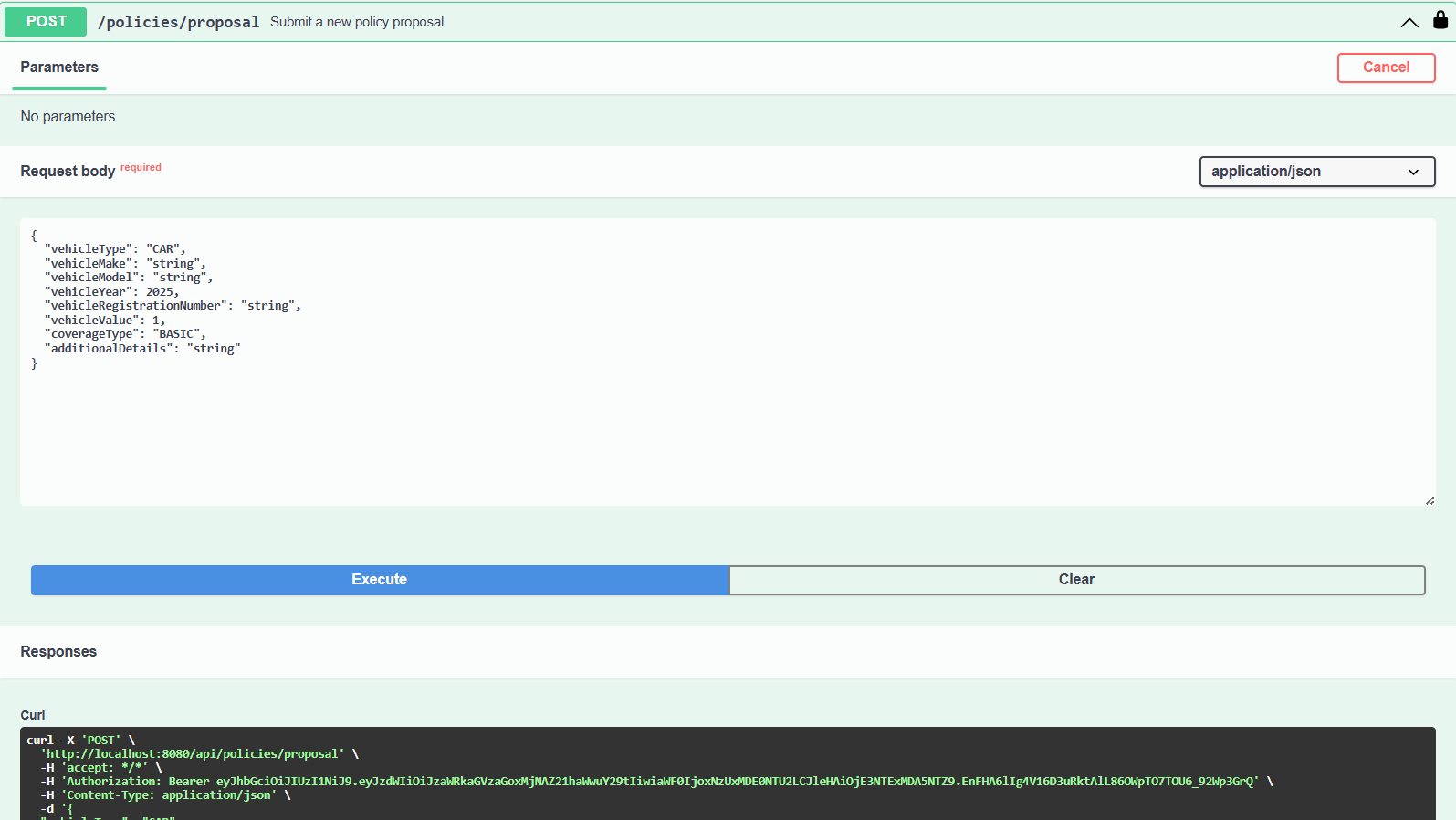
**1.3 Logout User**

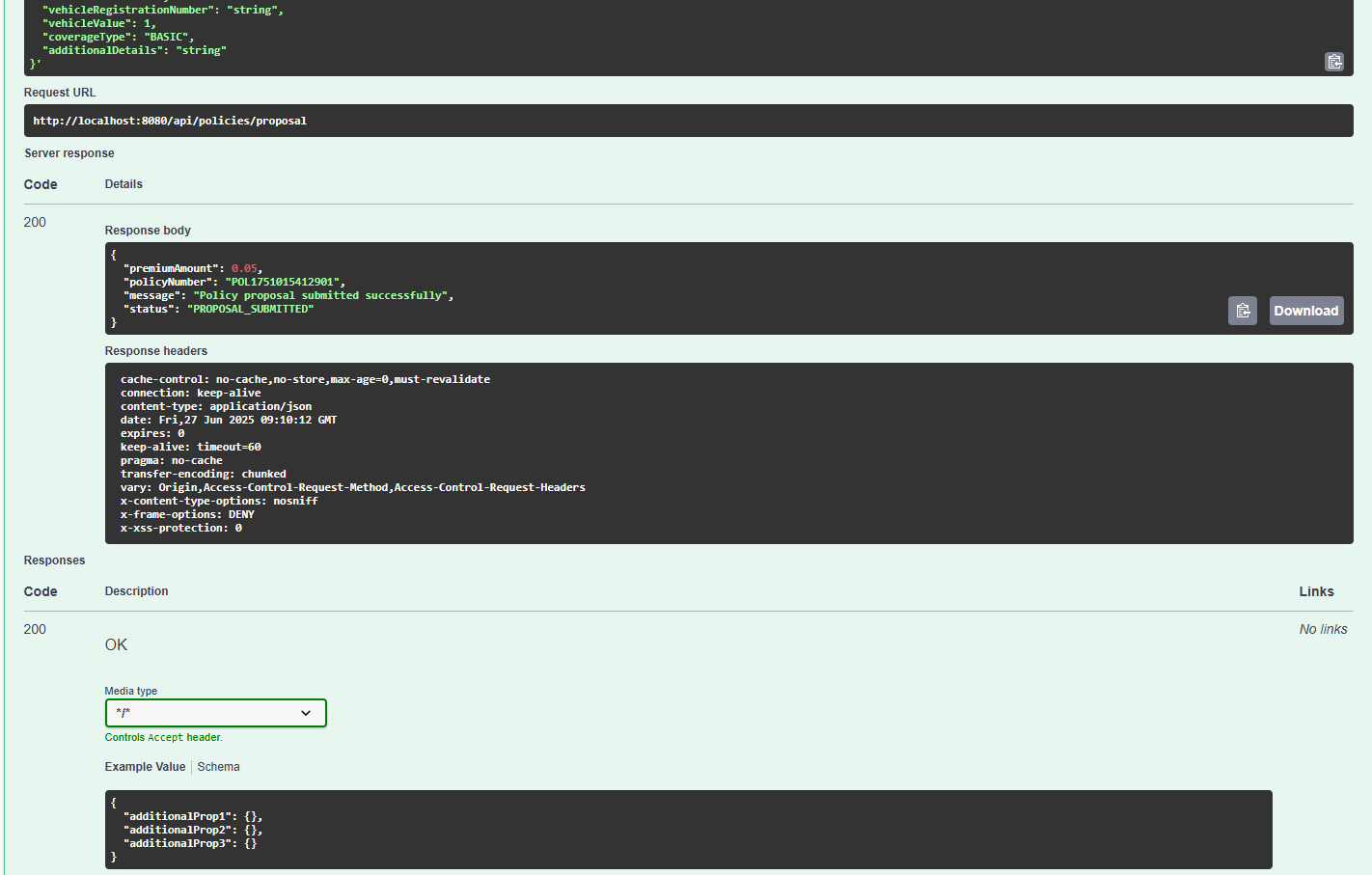
**POST /auth/logout**



**Phase 2: Policy Management (Available in PolicyController)**

**POST /policies/proposal**





{

"vehicleType": "CAR",

"vehicleMake": "Maruti Suzuki",

"vehicleModel": "Swift",

"vehicleYear": 2021,

"vehicleRegistrationNumber": "MH12AB1234",

"vehicleValue": 800000,

"coverageType": "COMPREHENSIVE",

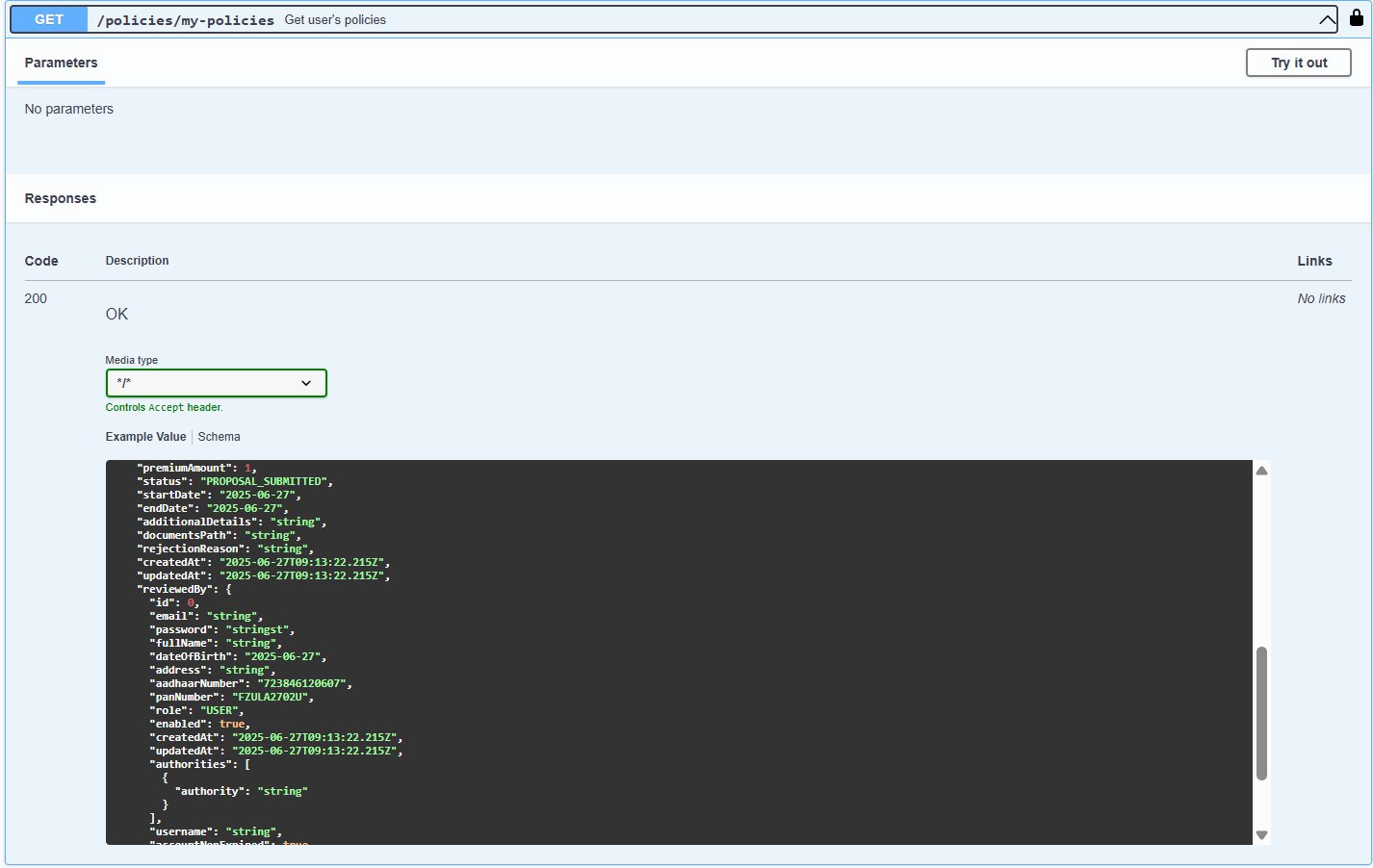
"additionalDetails": "Good condition, no previous accidents"

}

**2.2 Get My Policies**

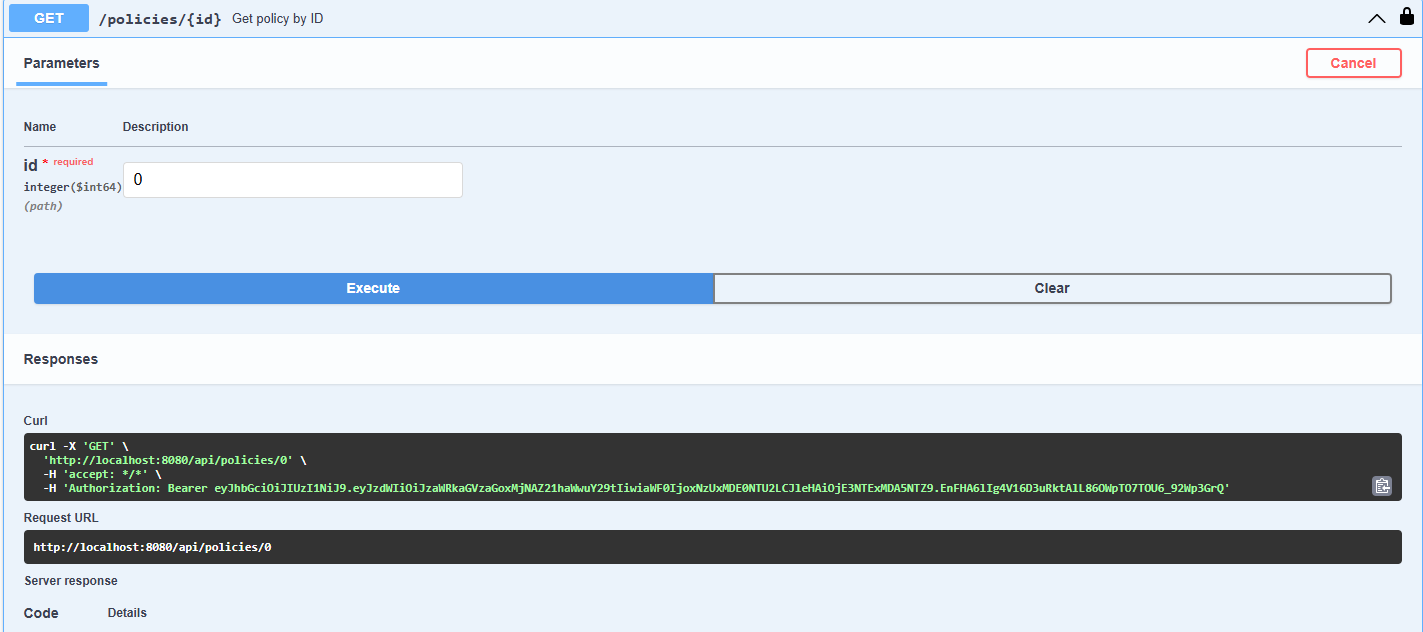
**GET /policies/my-policies**

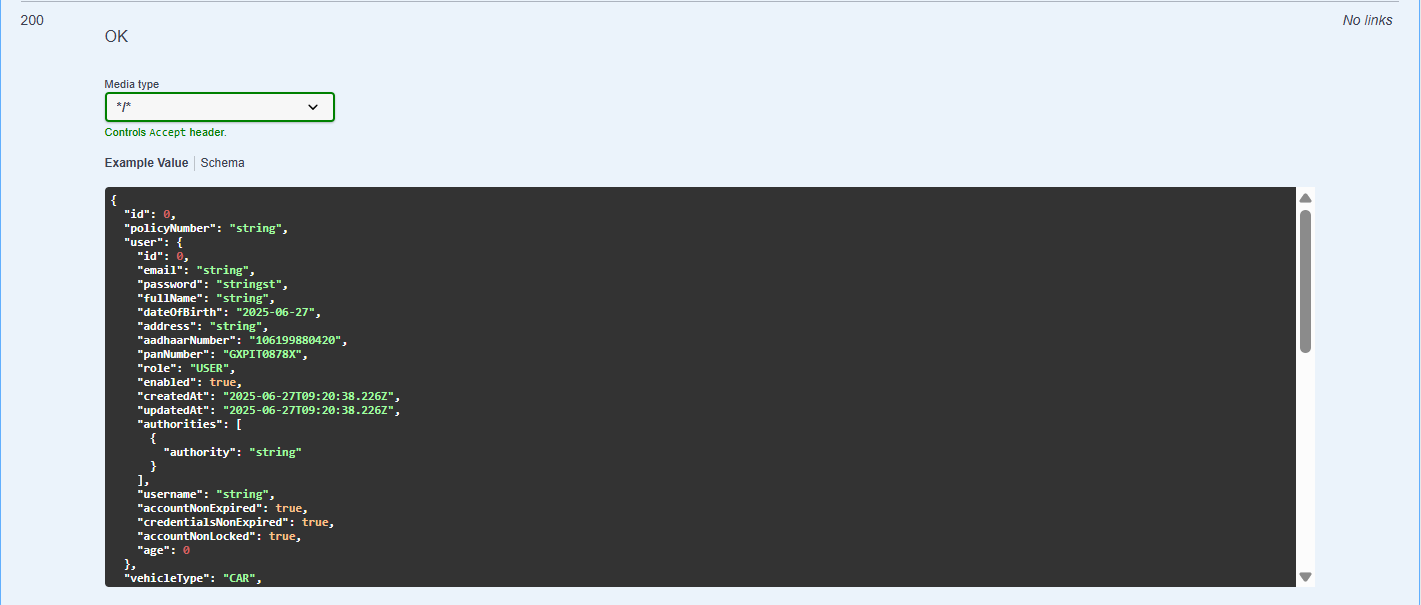
* Returns all policies for the authenticated user



**2.3 Get Policy by ID**

**GET /policies/{id}**

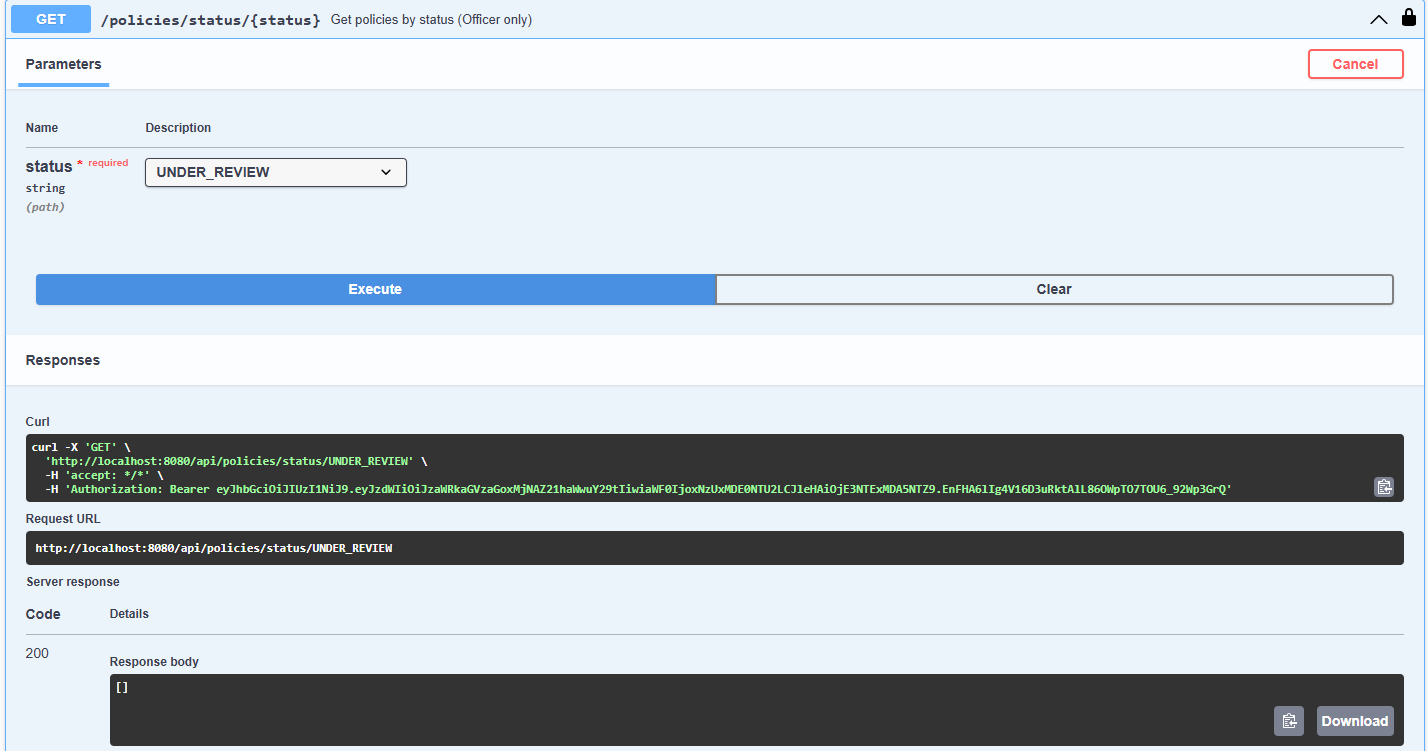


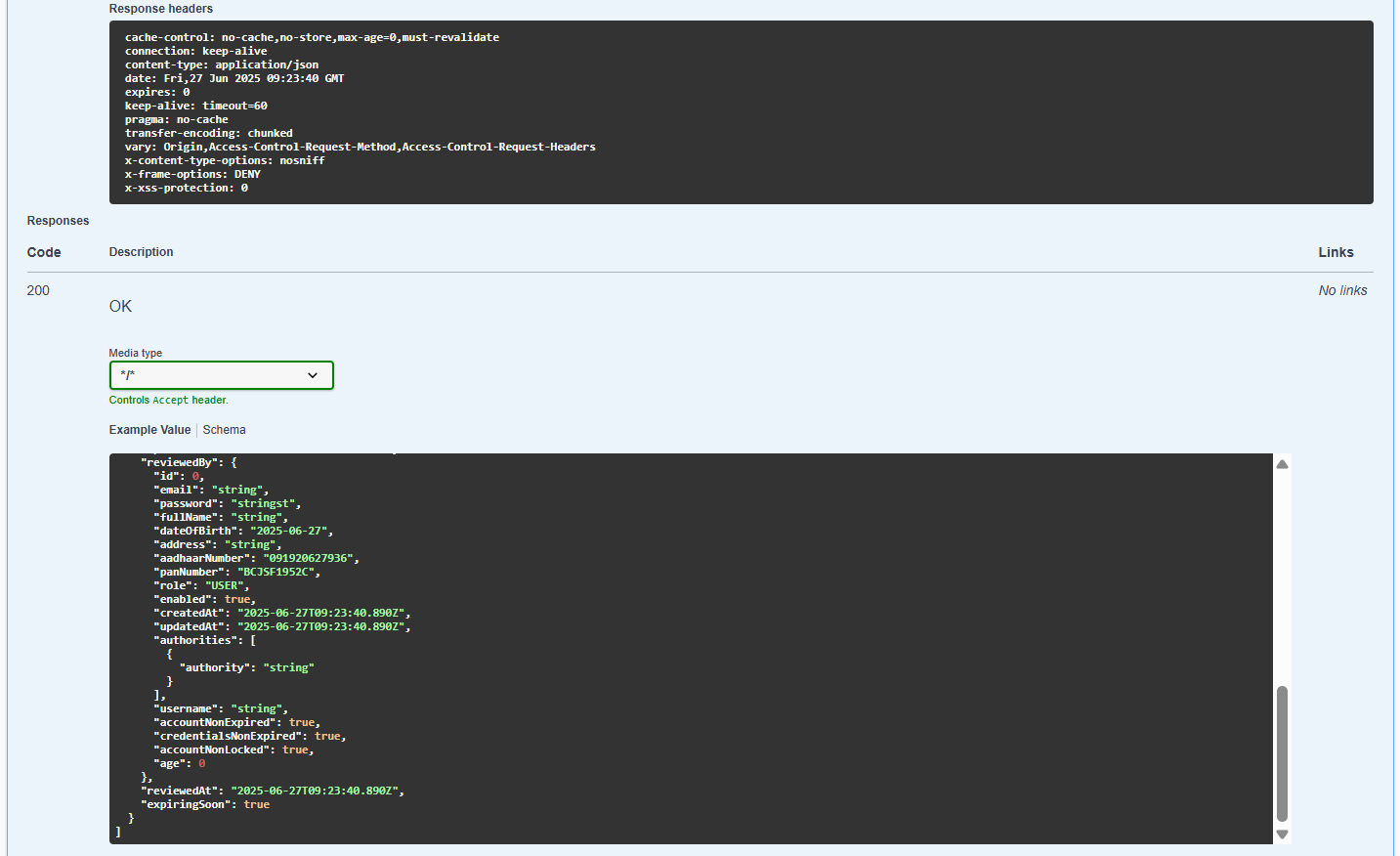


**2.4 Get Policies by Status**

**GET /policies/status/{status}**

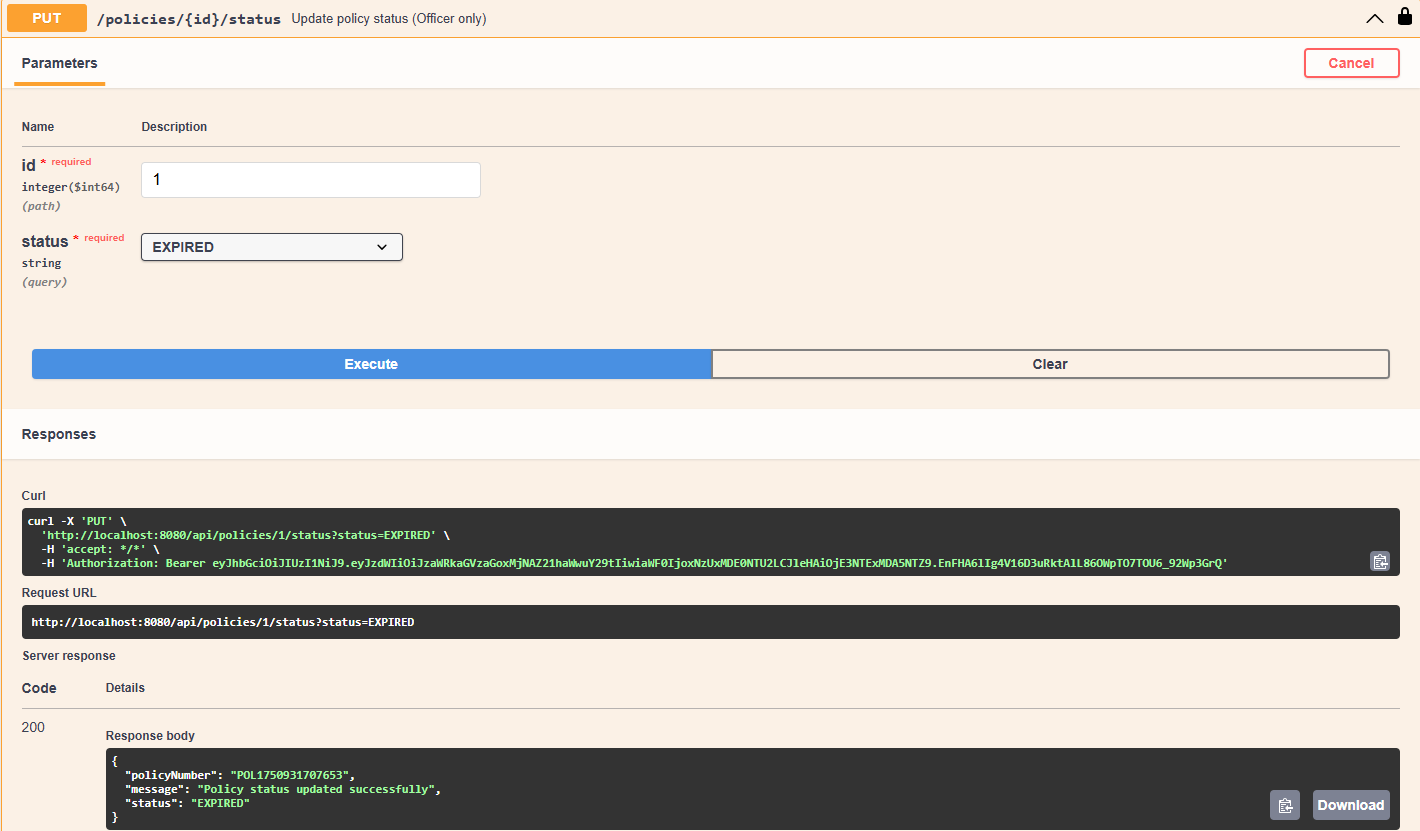
* Use status like: PROPOSAL, UNDER\_REVIEW, QUOTE, ACTIVE, EXPIRED

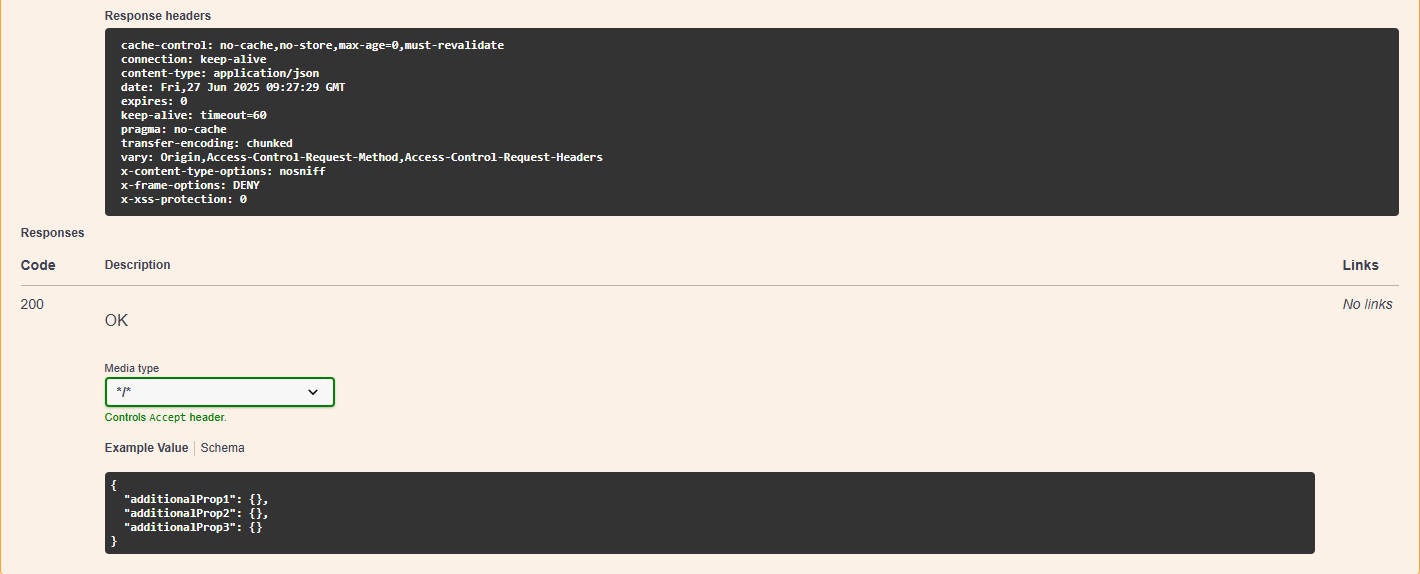




**2.5 Update Policy**

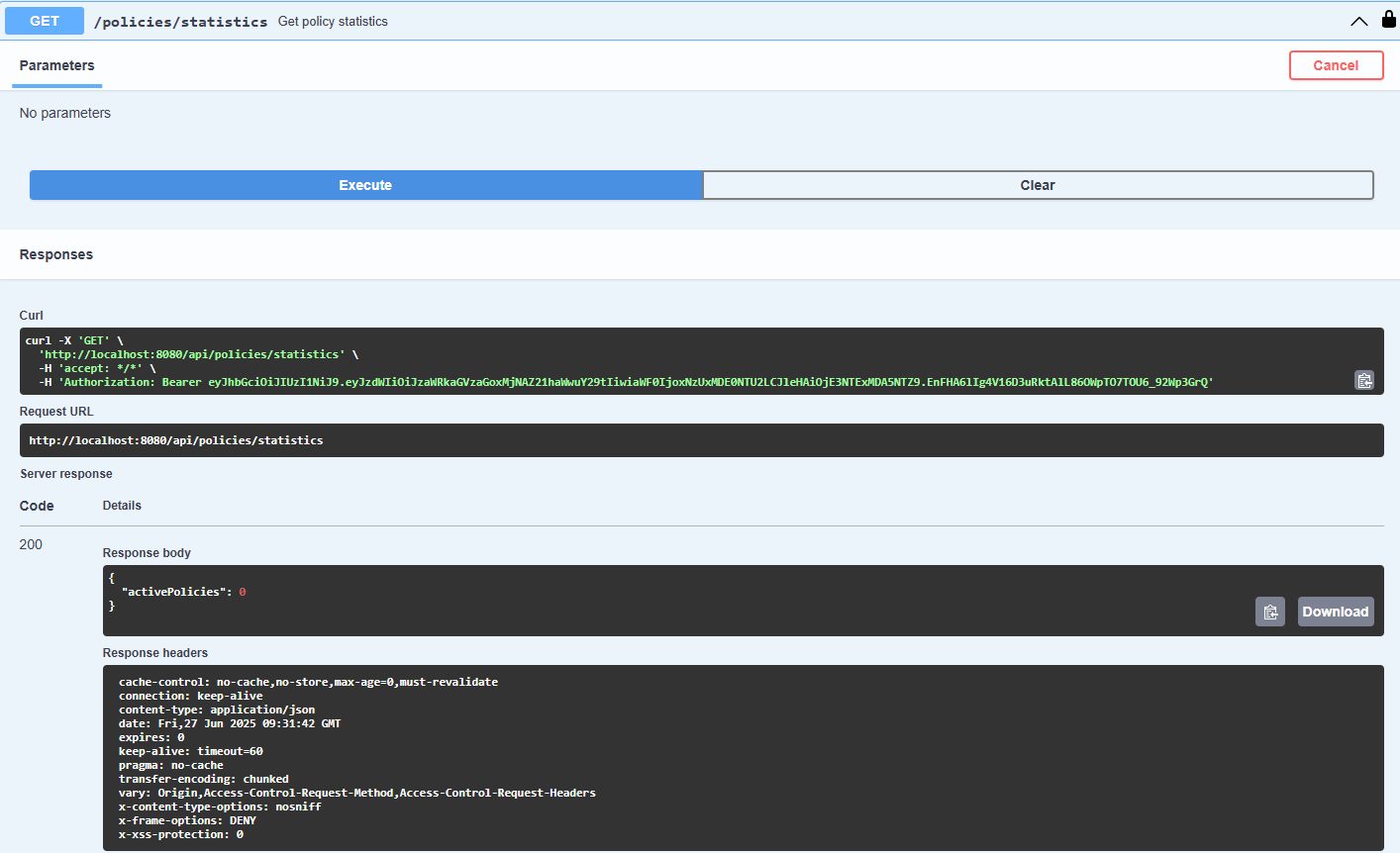
**PUT /policies/{id}**

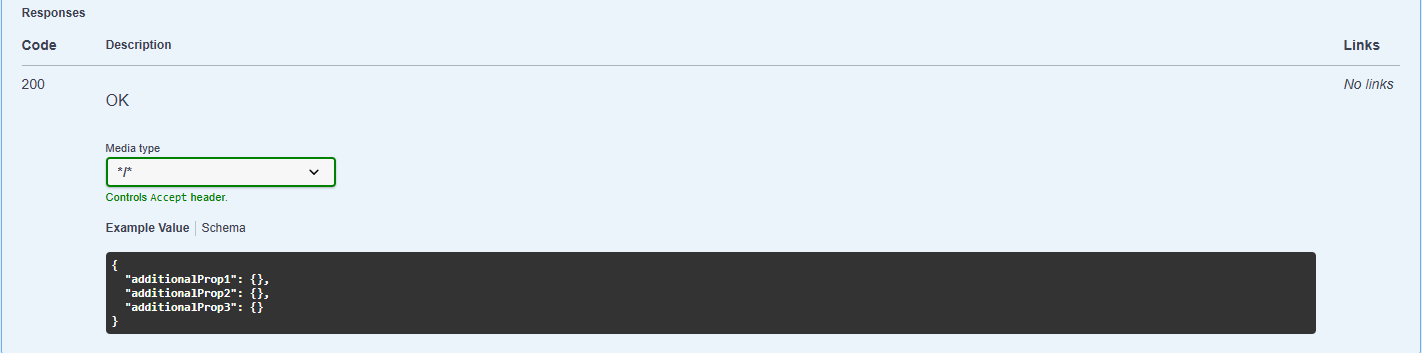


****

**2.6 Get Policy Statistics**

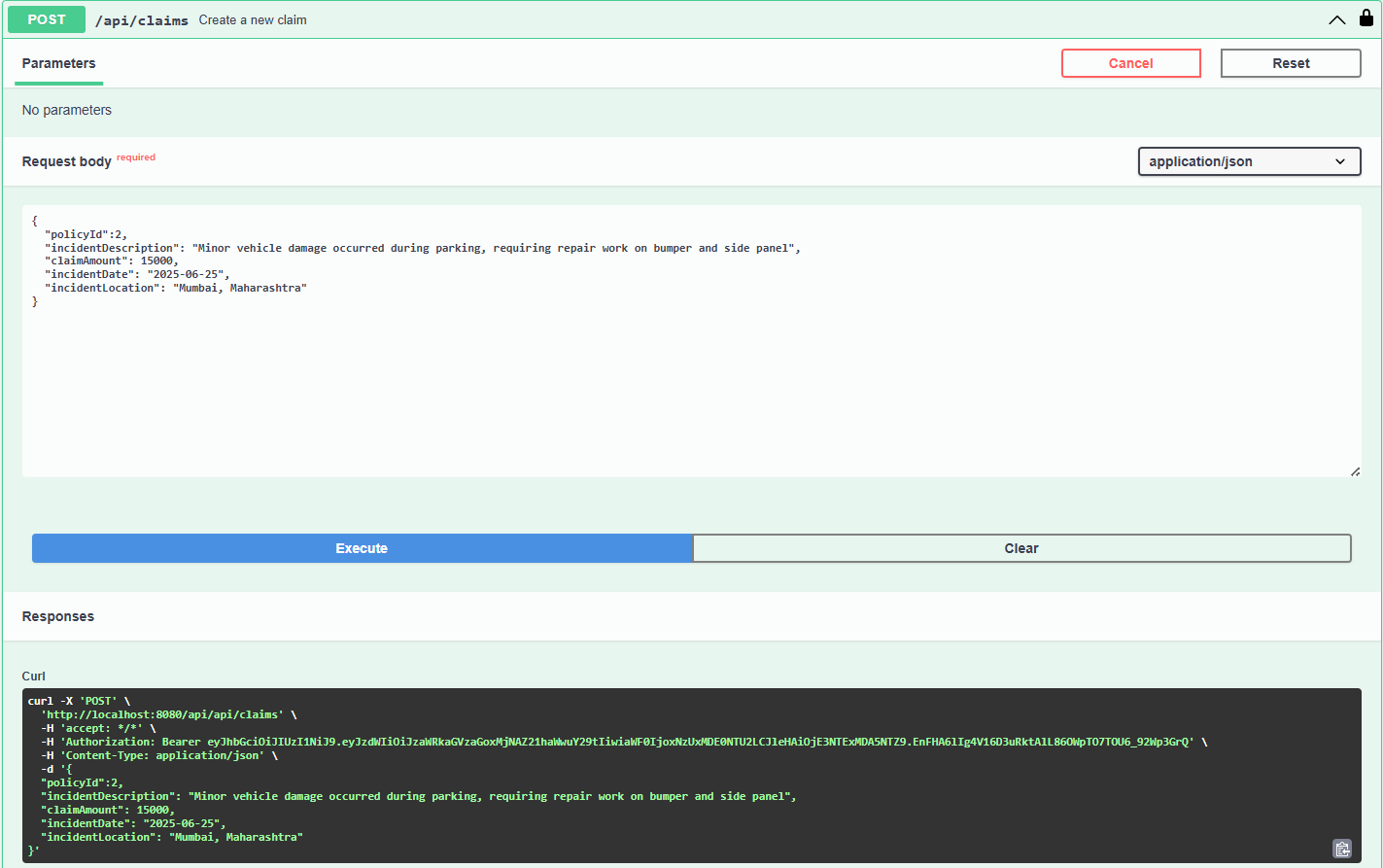
**GET /policies/statistics**

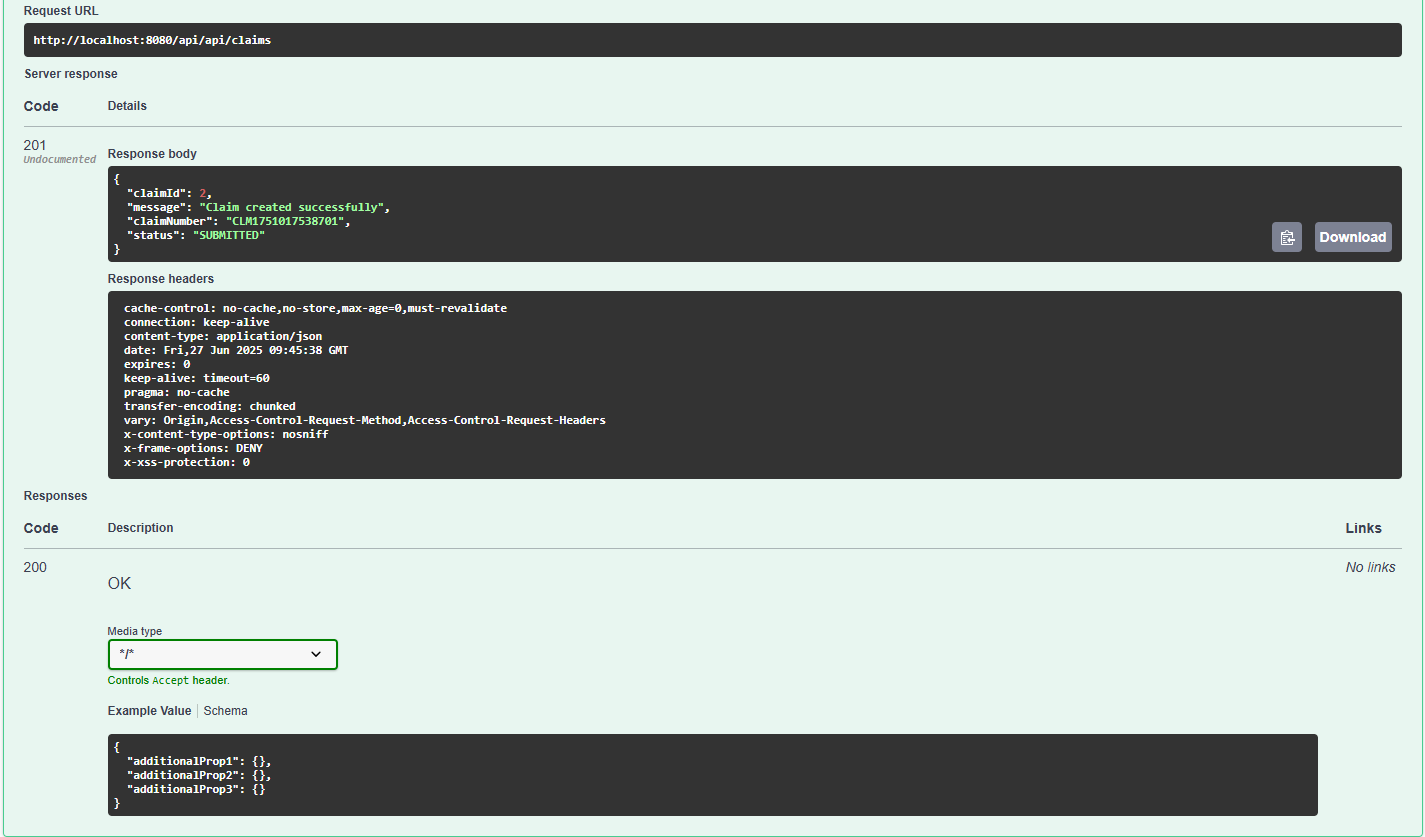




**3.1 Create Claim**

**POST /claims**

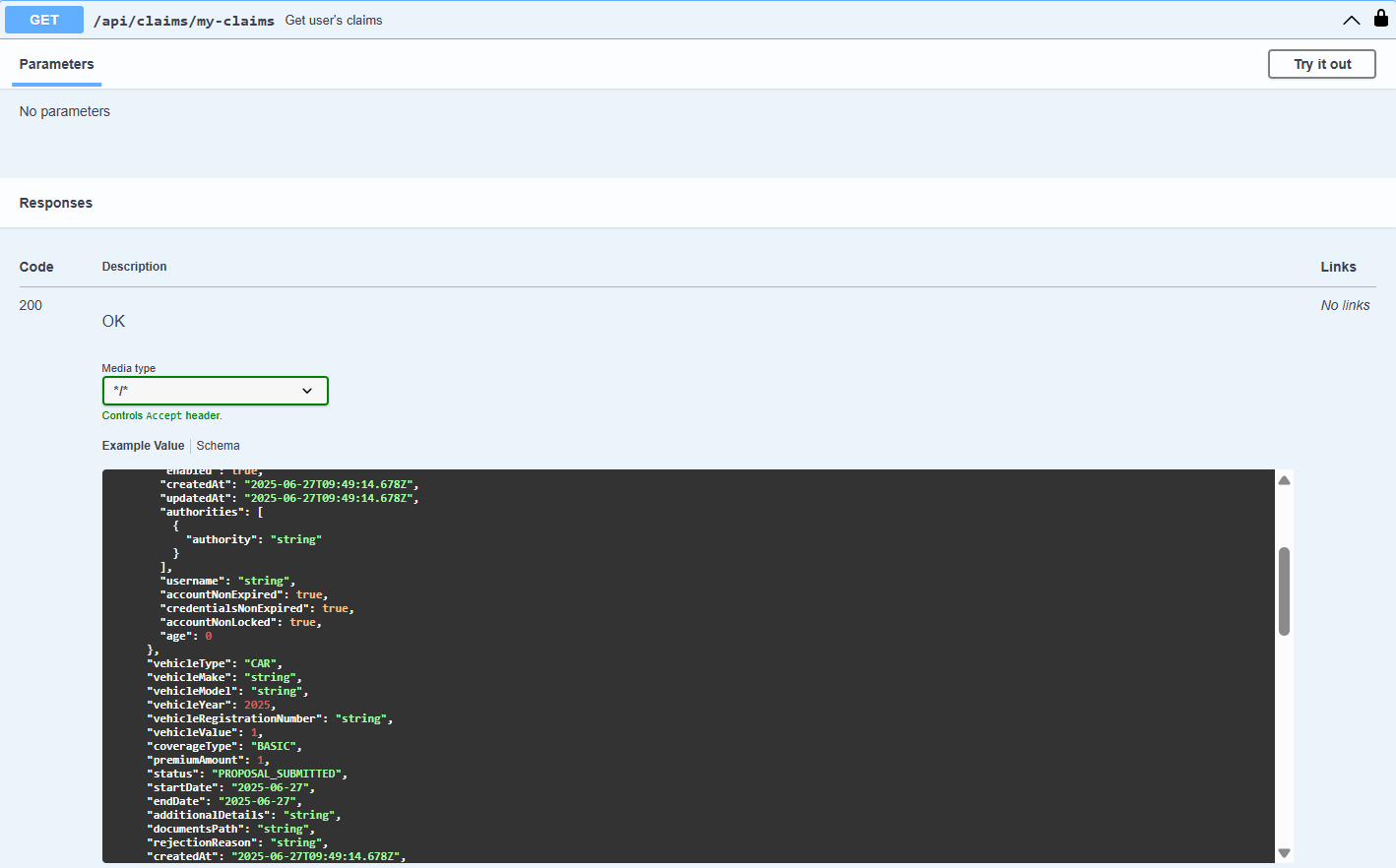




**3.2 Get My Claims**

**GET /claims/my-claims**

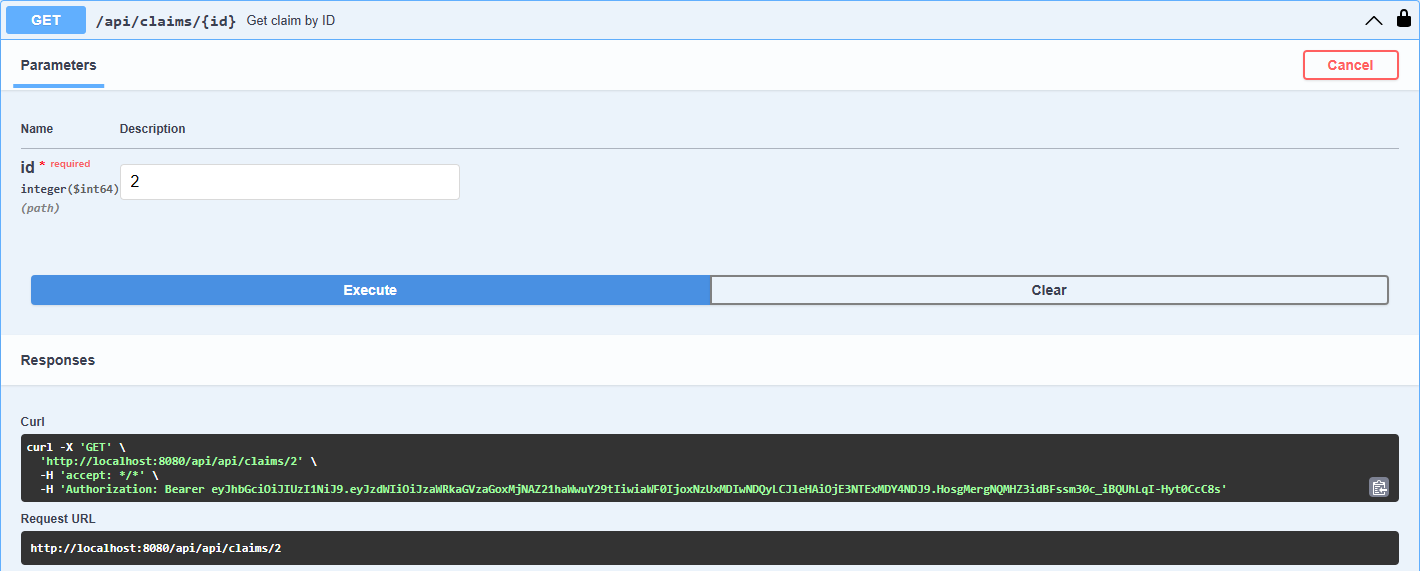
* Returns all claims for the authenticated user



**3.3 Get Claim by ID**

**GET /claims/2**

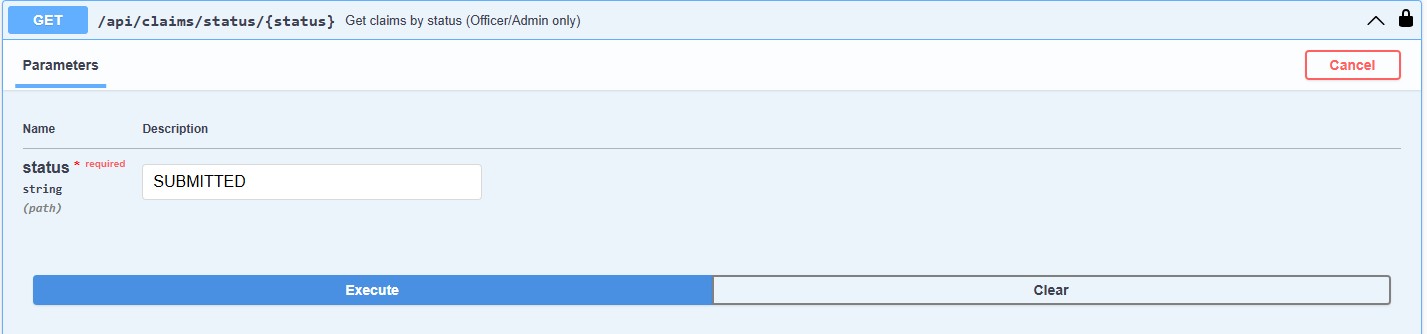
* Use the claim ID from your previous claim creation



**3.4 Get Claims by Status**

**GET /claims/status/SUBMITTED**

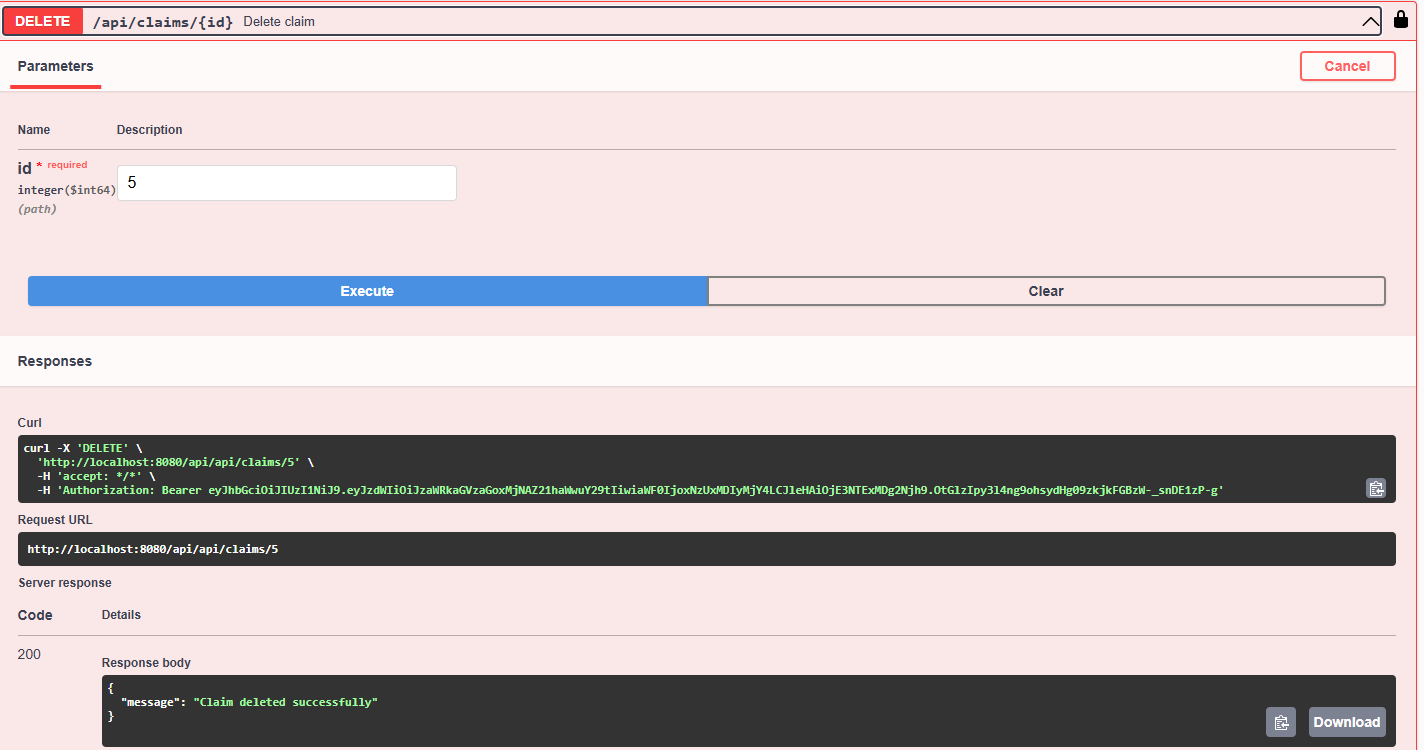
* Valid statuses: SUBMITTED, UNDER\_REVIEW, APPROVED, REJECTED, PAID



**3.4 Delete Claim**

**Delete/claims/{id}**

**You can only delete your own claims**

****

**Insurance Management System - Backend Folder Structure**

backend/

│

├── 📄 pom.xml # Maven configuration file

├── 📄 mvnw.cmd # Maven wrapper script (Windows)

│

├── 📁 .mvn/ # Maven wrapper configuration

│ └── wrapper/

│ └── maven-wrapper.properties

│

├── 📁 src/ # Source code directory

│ ├── 📁 main/ # Main application code

│ │ ├── 📁 java/ # Java source files

│ │ │ └── com/vehicleinsurance/ # Base package

│ │ │ │

│ │ │ ├── 📄 VehicleInsuranceApplication.java # Main Spring Boot class

│ │ │ │

│ │ │ ├── 📁 config/ # Configuration classes

│ │ │ │ └── SwaggerConfig.java

│ │ │ │

│ │ │ ├── 📁 controller/ # REST API controllers

│ │ │ │ ├── AuthController.java

│ │ │ │ ├── ClaimController.java

│ │ │ │ └── PolicyController.java

│ │ │ │

│ │ │ ├── 📁 dto/ # Data Transfer Objects

│ │ │ │ ├── ClaimDto.java

│ │ │ │ ├── LoginDto.java

│ │ │ │ ├── PolicyProposalDto.java

│ │ │ │ └── UserRegistrationDto.java

│ │ │ │

│ │ │ ├── 📁 entity/ # JPA Entity classes

│ │ │ │ ├── Claim.java

│ │ │ │ ├── ClaimStatus.java

│ │ │ │ ├── CoverageType.java

│ │ │ │ ├── Policy.java

│ │ │ │ ├── PolicyStatus.java

│ │ │ │ ├── Role.java

│ │ │ │ ├── User.java

│ │ │ │ └── VehicleType.java

│ │ │ │

│ │ │ ├── 📁 repository/ # JPA Repository interfaces

│ │ │ │ ├── ClaimRepository.java

│ │ │ │ ├── PolicyRepository.java

│ │ │ │ └── UserRepository.java

│ │ │ │

│ │ │ ├── 📁 security/ # Security configuration

│ │ │ │ ├── AuthEntryPointJwt.java

│ │ │ │ ├── AuthTokenFilter.java

│ │ │ │ ├── JwtUtils.java

│ │ │ │ ├── UserDetailsServiceImpl.java

│ │ │ │ └── WebSecurityConfig.java

│ │ │ │

│ │ │ └── 📁 service/ # Business logic services

│ │ │ ├── ClaimService.java

│ │ │ ├── PolicyService.java

│ │ │ ├── PremiumCalculationService.java

│ │ │ └── UserService.java

│ │ │

│ │ └── 📁 resources/ # Application resources

│ │ └── application.properties # Spring Boot configuration

│ │

│ └── 📁 test/ # Test source files

│ └── java/

│ └── com/vehicleinsurance/ # Test classes mirror structure

│

└── 📁 target/ # Build output directory (generated)

├── 📄 vehicle-insurance-backend-1.0.0.jar # Executable JAR file

├── 📁 classes/ # Compiled Java classes

├── 📁 generated-sources/ # Generated source files

├── 📁 maven-archiver/ # Maven build metadata

├── 📁 maven-status/ # Maven compilation status

└── 📁 test-classes/ # Compiled test classes

**Detailed Package Breakdown:**

**🎯 Controller Layer (**[controller/](vscode-file://vscode-app/c:/Users/vsour/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**):**

* [AuthController.java](vscode-file://vscode-app/c:/Users/vsour/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) - Authentication endpoints (register, login)
* [ClaimController.java](vscode-file://vscode-app/c:/Users/vsour/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) - Claim management endpoints
* [PolicyController.java](vscode-file://vscode-app/c:/Users/vsour/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) - Policy management endpoints

**📦 DTO Layer (**[dto/](vscode-file://vscode-app/c:/Users/vsour/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ")**):**

* **ClaimDto.java** - Claim data transfer object
* **LoginDto.java** - Login request/response
* **PolicyProposalDto.java** - Policy creation data
* **UserRegistrationDto.java** - User registration data

**🗄️ Entity Layer (**[entity/](vscode-file://vscode-app/c:/Users/vsour/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**):**

* **User.java** - User entity with roles
* **Policy.java** - Insurance policy entity
* **Claim.java** - Insurance claim entity
* **Enums:** ClaimStatus, CoverageType, PolicyStatus, Role, VehicleType

**🔗 Repository Layer (**[repository/](vscode-file://vscode-app/c:/Users/vsour/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**):**

* **UserRepository.java** - User data access
* **PolicyRepository.java** - Policy data access
* [ClaimRepository.java](vscode-file://vscode-app/c:/Users/vsour/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) - Claim data access

**🔐 Security Layer (**[security/](vscode-file://vscode-app/c:/Users/vsour/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**):**

* **WebSecurityConfig.java** - Main security configuration
* **JwtUtils.java** - JWT token utilities
* **AuthTokenFilter.java** - JWT authentication filter
* **UserDetailsServiceImpl.java** - User details service
* **AuthEntryPointJwt.java** - Unauthorized access handler

**⚙️ Service Layer (**[service/](vscode-file://vscode-app/c:/Users/vsour/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**):**

* **UserService.java** - User business logic
* **PolicyService.java** - Policy business logic
* [ClaimService.java](vscode-file://vscode-app/c:/Users/vsour/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) - Claim business logic
* **PremiumCalculationService.java** - Premium calculation logic

**🔧 Configuration (config/):**

* **SwaggerConfig.java** - API documentation configuration

**📋 Key Files:**

* **pom.xml** - Maven dependencies and build configuration
* [application.properties](vscode-file://vscode-app/c:/Users/vsour/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) - Database, JWT, and application settings
* **VehicleInsuranceApplication.java** - Spring Boot main class