Customizing Swagger for Your API

Craig Golightly
SENIOR SOFTWARE CONSULTANT

@seethatgo   www.seethatgo.com

Spring is opinionated

Save most time when follow conventions

Springfox leverages Spring opinions

- Saves you from annotating everything

Maximize what framework can do for you

```
public class Employee {

  @ApiModelProperty(value="First and last name")
  public String name;
```

- Only use if you have something to add that is not obvious from the naming

```
public class Employee {

  ~~@ApiModelProperty(value="Employee department")~~
  public String department;
```

- Classname + variable name is NOT useful information

```
import io.swagger.annotations.ApiModelProperty;
```

- Annotation package

Demo

Add @ApiModelProperty annotation

View in the Swagger UI

```
@NotNull, @Min, @Max, @Size
```

- Automatically translate JSR-303 annotations in documentation

```
<dependency>
 <groupId>io.springfox</groupId>
 <artifactId>springfox-bean-validators</artifactId>
 <version>${io.springfox.version}</version>
</dependency>
```

- Dependency

```
@Import(BeanValidatorPluginsConfiguration.class)
```

- Import on configuration class

```
@ApiModelProperty(value = "Description“)
@NotNull
public String department;
```

- ApiModelProperty takes prescedence

```
@ApiModelProperty(value = "Description",
required = true)
@NotNull
public String department;
```

- Must add required=true

```java
@ApiOperation(value="Get the default schedule for employees",
        notes="All employees share the same schedule.")
@RequestMapping(value = "/schedule", method = RequestMethod.GET)
public String getSchedule() {

    return "Your schedule is M-F 9-5";
}
```

Swagger Annotations

@ApiOperation

- Value
- Notes

Demo

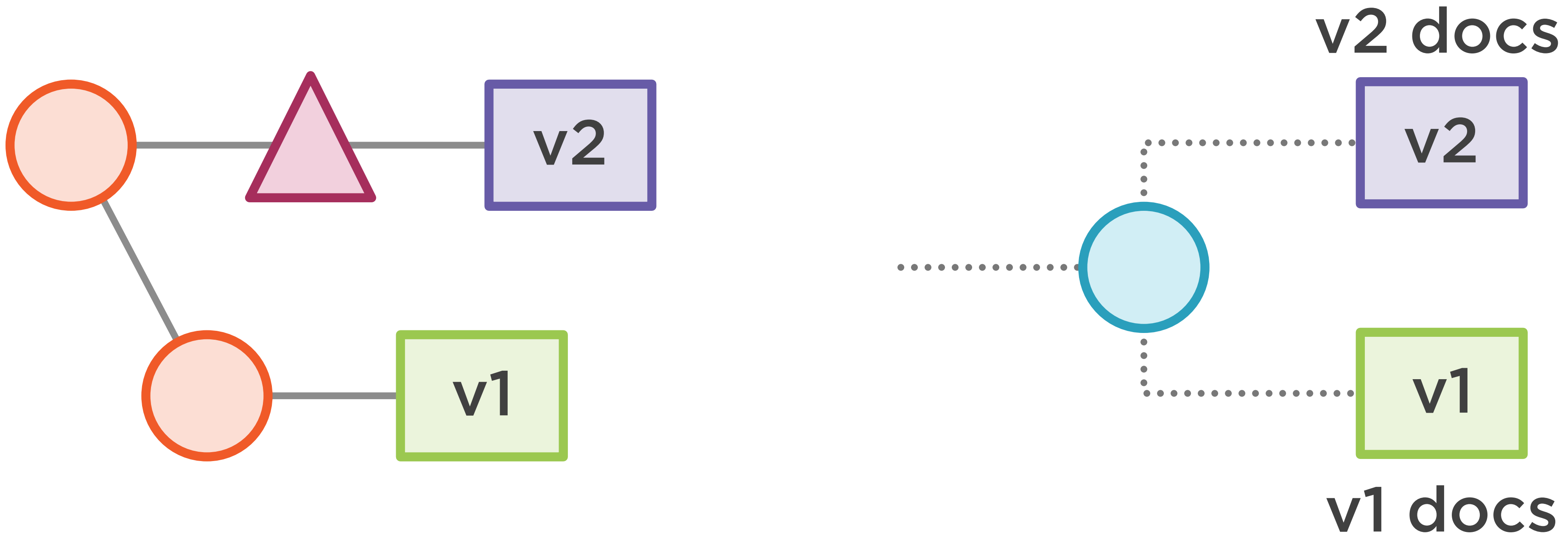
Add bean validators

Observe precedence of @ApiModelProperty

Add required property

@ApiOperation

Branching and Load Balancer



```
new Docket(DocumentationType.SWAGGER_2)
        .select()
        .apis(RequestHandlerSelectors.any())
        .paths(PathSelectors.ant("/v2/**"))
        .build()
        .apiInfo(getApiInfo());
```

PathSelectors

In Docket Object

Restrict documentation to certain endpoints

- Based on path

Demo

Two version of API

- Each with own Swagger UI

PathSelectors

```
new Docket(DocumentationType.SWAGGER_2)
        .groupName(“employee-api-v2")
        .select()
        .apis(RequestHandlerSelectors.any())
        .paths(PathSelectors.ant("/v2/**"))
        .build()
        .apiInfo(getApiInfo());
```

Swagger Groups

Multiple versions of documentation from single Swagger UI

Selector on Swagger UI for group

PathSelectors can filter each version into different Docket Objects and groups

Demo

Swagger groups

- Multiple Docket Objects
- Switch groups in the Swagger UI

Summary

Leverage annotations for documentation

- Bean validator for JSR-303
- @ApiModelProperty
- @ApiOperation

Documenting different versions of API

- PathSelectors
- Swagger groups

API Documentation based on code

- Easy to keep up to date