

# MicroProfile OpenAPI

This cheat sheet covers the basics of MicroProfile OpenAPI specification uses.

## THE BASICS

### Usage

`/openapi` -- base endpoint for OpenAPI document

### Annotations

- `@Callback` -- a callback URL that will be invoked
- `@Callbacks` -- an array of callbacks URLs
- `@CallbackOperation` -- operation that will be invoked during the callback
- `@Components` -- container for various reusable objects
- `@Explode` -- enumeration for the `explode` property
- `@ParameterIn` -- enumeration for the `in` property
- `@ParamterStyle` -- enumeration for the `style` property
- `@SecuritySchemeIn` -- enumeration for the security scheme's `in` property
- `@SecuritySchemeType` -- enumeration for the security scheme's `type` property
- `@Extension` -- add an extension with contained properties
- `@Extensions` -- add custom properties to an extension
- `@ExternalDocumentation` -- an external resource for extended documentation
- `@Header` -- a header object
- `@Contact` -- contact information for the exposed API
- `@Info` -- metadata about the API
- `@License` -- license of the exposed API
- `@Link` -- design-time link for a response
- `@LinkParameter` -- parameter to pass to the linked operation
- `@Content` -- schema and examples for a particular media type
- `@DiscriminatorMapping` -- differentiation between schemas
- `@Encoding` -- encoding definition applied to a single Schema object
- `@ExampleObject` -- example of a particular content
- `@Schema` -- the definition of input and output data types
- `@OpenAPIDefinition` -- general metadata for an OpenAPI definition
- `@Operation` -- operation (e.g. HTTP method) against a specific path
- `@Parameter` -- single operation parameter
- `@Parameters` -- encapsulates multiple parameters
- `@RequestBody` -- single request body
- `@APIResponse` -- single response from an API operation
- `@APIResponses` -- container for multiple API responses
- `@OAuthFlow` -- configuration detail for a supported OAuth Flow
- `@OAuthFlows` -- configuration for multiple OAuth Flows
- `@OAuthScope` -- represents OAuth scope
- `@SecurityRequirement` -- security requirement for an operation
- `@SecurityRequirements` -- multiple security requirements (only one needs to be satisfied)
- `@SecurityRequirementsSet` -- multiple security requirement (all need to be satisfied)
- `@SecurityScheme` -- defines a security scheme
- `@SecuritySchemes` -- defines an array of security schemes
- `@Server` -- represents a server used in operations
- `@Servers` -- a container of multiple server definitions

**@ServerVariable** -- a server variable for server URL template substitution

**@Tag** -- a tag for the API endpoint

**@Tags** -- a container of multiple tags

### Example

```
@GET
@Path("/findByStatus")
@Operation(summary = "Finds Pets by status",
    description = "Multiple status values
    can be provided with comma separated
    strings")
public Response findPetsByStatus(...) { ... }

@GET
@Path("/{username}")
@Operation(summary = "Get user by user name")
@ApiResponse(description = "The user",
    content = @Content(mediaType =
        "application/json",
            schema =
                @Schema(implementation
                    = User.class))),
@ApiResponse(responseCode = "400", description =
    "User not found")
public Response getUserByName(
    @Parameter(description = "The name that
        needs to be fetched. Use user1
        for testing. ", required = true)
    @PathParam("username") String username)
{ ... }
```

### Programming model

**OASFactory** -- create all of the elements of an OpenAPI tree

```
OASFactory.createObject(Info.class).title("Airlines")
    .description("Airlines APIs").version("1.0.0");
```

**OASModelReader** -- bootstrap the OpenAPI model tree used by the processing framework  
registration:

```
mp.openapi.model.reader=com.mypackage.MyModelReader
```

**OASFilter** -- receive callbacks for various key OpenAPI elements  
registration:

```
mp.openapi.filter=com.mypackage.MyFilter
```

### Configuration – core values

**@mp.openapi.model.reader** -- full name of OASModelReader implementation

**@mp.openapi.filter** -- full name of OASFilter implementation

**@mp.openapi.scan.disable** -- disable annotation scanning

**@mp.openapi.scan.packages** -- specify packages to scan

**@mp.openapi.scan.classes** -- specify classes to scan

**@mp.openapi.scan.exclude.packages** -- specify packages excluded from scanning

**@mp.openapi.scan.exclude.classes** -- specify classes excluded from scanning

**@mp.openapi.servers** -- specify list of global servers

**@mp.openapi.servers.path.** -- prefix of config properties to specify alternative list of servers in a path

**@mp.openapi.servers.operation.** -- prefix of config properties to specify alternative list of servers to service an operation

**Author** Martin Stefanko