

Университет ИТМО»

Факультет ПИиКТ

Дисциплина: Технологии веб-сервисов

Лабораторная работа 5

Задача

Выполнение

```
/**
 * Creates a city.
 *
 * @param city creation info DTO.
 * @return created city.
 */
@POST
@Path("/")
@Consumes(MediaType.APPLICATION_JSON)
@SneakyThrows
@Operation(
    summary = "Creates a city",
    tags = {"city"},
    description = "Create a city",
    responses = {
        @ApiResponse(
            description = "The city",
            content = @Content(schema = @Schema(implementation =
CityDTO.class))),
        @ApiResponse(responseCode = "400", description = "Invalid
request data"),
    })
public CityDTO create(
    @RequestBody(
        description = "City to update",
        required = true,
        content = @Content(schema = @Schema(implementation
= CityCreateUpdateDTO.class)))
        final CityCreateUpdateDTO city) {
    return this.cityService.create(city);
}

/**
 * Update city by id with full-data entity.
 *
 * @param id city identifier.
 * @param city new city data.
 * @return updated city.
```

```

    */
    @PUT
    @Path("/{id}")
    @Consumes(MediaType.APPLICATION_JSON)
    @SneakyThrows
    @Operation(
        summary = "Update the city",
        tags = {"city"},
        description = "Update the city with full schema",
        responses = {
            @ApiResponse(
                description = "The city",
                content = @Content(schema = @Schema(implementation =
CityDTO.class))),
            @ApiResponse(responseCode = "404", description = "City not
found"),
            @ApiResponse(responseCode = "400", description = "Invalid
request data"),
        })
    public CityDTO update(
        @Parameter(description = "City id", required = true)
        @PathParam("id") final Integer id,
        @RequestBody(
            description = "City to update",
            required = true,
            content = @Content(schema = @Schema(implementation
= CityCreateUpdatedDTO.class)))
        final CityCreateUpdatedDTO city) {
        return this.cityService.update(id, city);
    }

    /**
     * Update part of city data. Updates only non-empty fields.
     *
     * @param id city identifier.
     * @param city city update data.
     * @return updated city.
     */
    @PATCH
    @Path("/{id}")
    @Consumes(MediaType.APPLICATION_JSON)
    @SneakyThrows
    @Operation(
        summary = "Patch the city",
        tags = {"city"},
        description = "Patch the city partially",
        responses = {
            @ApiResponse(
                description = "The city",
                content = @Content(schema = @Schema(implementation =
CityDTO.class))),
            @ApiResponse(responseCode = "404", description = "City not
found"),
            @ApiResponse(responseCode = "400", description = "Invalid
request data"),
        })

```

```

public CityDTO patch(
    @Parameter(description = "City id", required = true)
    @PathParam("id") final Integer id,
    @RequestBody(
        description = "City to patch",
        required = true,
        content = @Content(schema = @Schema(implementation
= CityCreateUpdatedDTO.class)))
    final CityCreateUpdatedDTO city) {
    return this.cityService.patch(id, city);
}

/**
 * Delete city by id.
 *
 * @param id The city identifier.
 * @return OK if deleted.
 */
@DELETE
@Path("/{id}")
@SneakyThrows
@Operation(
    summary = "Delete the city",
    tags = {"city"},
    description = "Delete the city",
    responses = {
        @ApiResponse(description = "The city deletion status"),
        @ApiResponse(responseCode = "404", description = "City not
found"),
        @ApiResponse(responseCode = "400", description = "Invalid
request data"),
    })
public Response delete(@Parameter(description = "City id", required = true)
@PathParam("id") final Integer id) {
    this.cityService.deleteById(id);
    return Response.ok().build();
}

```

The image shows the Swagger UI interface for an API. At the top, the Swagger logo is visible, along with the URL `http://localhost:9596/lab5-server/api/openapi.json` and an `Explore` button. Below the URL bar, there is a `Servers` dropdown menu currently set to `/lab5-server`. The main content area displays a list of API endpoints under the `city` resource. The endpoints are:

- GET** `/api/city`: Find cities by filters
- POST** `/api/city`: Creates a city
- PUT** `/api/city/{id}`: Update the city
- DELETE** `/api/city/{id}`: Delete the city
- PATCH** `/api/city/{id}`: Patch the city

Each endpoint is represented by a colored bar with its HTTP method, path, and a brief description. The `DELETE` method is highlighted in red, while the others are in various shades of blue, green, and orange.

Вывод