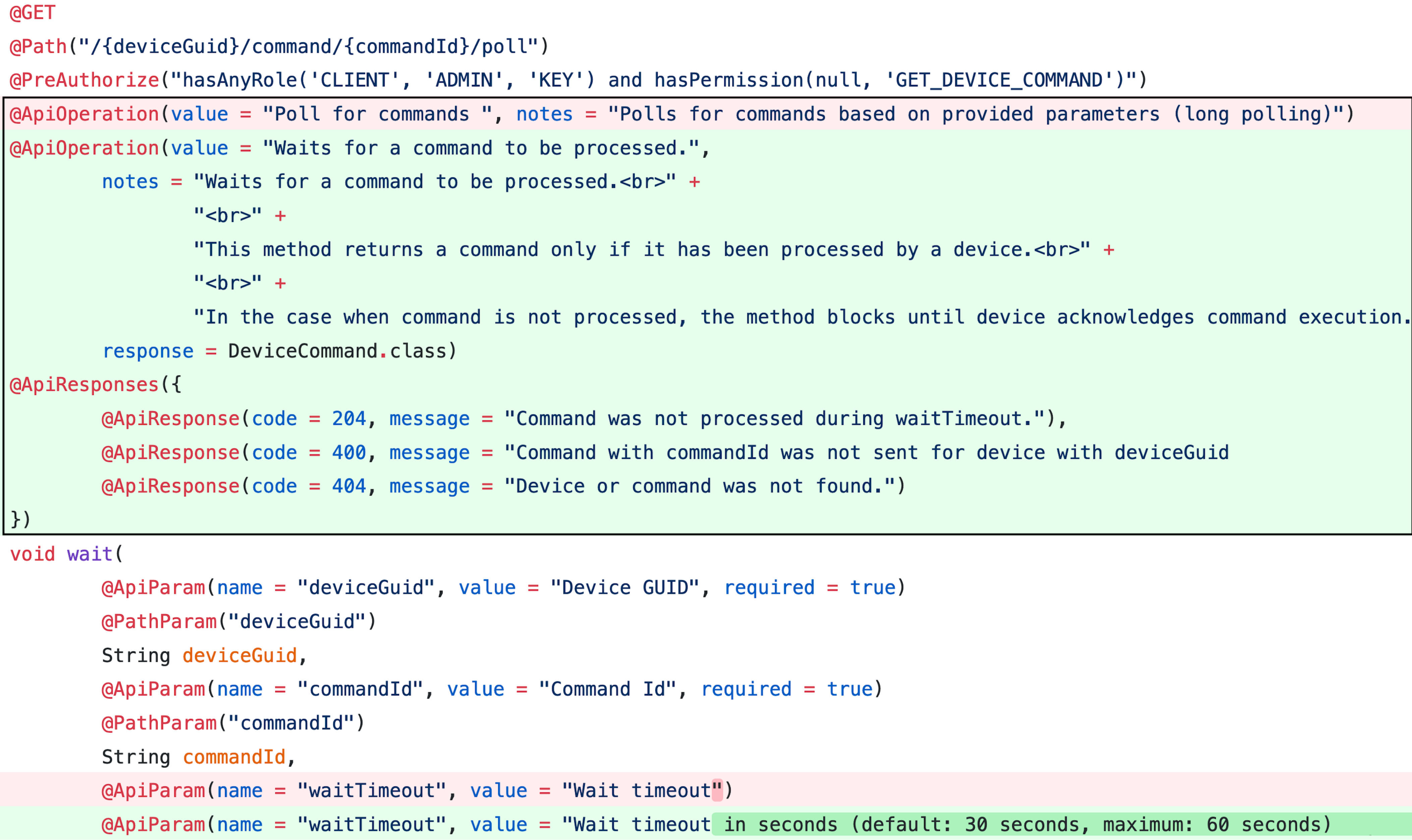

```
@GET
@Path("/{deviceGuid}/command/{commandId}/poll")
@PreAuthorize("hasAnyRole('CLIENT', 'ADMIN', 'KEY') and hasPermission(null, 'GET_DEVICE_COMMAND')")
@ApiOperation(value = "Poll for commands ", notes = "Polls for commands based on provided parameters (long polling)")
@ApiOperation(value = "Waits for a command to be processed.",
        notes = "Waits for a command to be processed.<br>" +
                "<br>" +
                "This method returns a command only if it has been processed by a device.<br>" +
                "<br>" +
                "In the case when command is not processed, the method blocks until device acknowledges command execution.
        response = DeviceCommand.class)
@ApiResponses({
        @ApiResponse(code = 204, message = "Command was not processed during waitTimeout."),
        @ApiResponse(code = 400, message = "Command with commandId was not sent for device with deviceGuid
        @ApiResponse(code = 404, message = "Device or command was not found.")
})
void wait(
        @ApiParam(name = "deviceGuid", value = "Device GUID", required = true)
        @PathParam("deviceGuid")
        String deviceGuid,
        @ApiParam(name = "commandId", value = "Command Id", required = true)
        @PathParam("commandId")
        String commandId,
        @ApiParam(name = "waitTimeout", value = "Wait timeout")
        @ApiParam(name = "waitTimeout", value = "Wait timeout in seconds (default: 30 seconds, maximum: 60 seconds)
```