

Pretty-print ☒

```

{
  "openapi": "3.0.1",
  "info": {
    "title": "Robot Controller API",
    "description": "New backend service that provides resources for the Moon robot simulator.",
    "contact": {
      "name": "Prerna Shahi",
      "email": "s222486984@deakin.edu.au"
    }
  },
  "paths": {
    "/api/maps": {
      "get": {
        "tags": [
          "Maps"
        ],
        "summary": "Retrieves all maps.",
        "responses": {
          "200": {
            "description": "Returns Ok with the list of maps",
            "content": {
              "text/plain": {
                "schema": {
                  "type": "array",
                  "items": {
                    "$ref": "#/components/schemas/Map"
                  }
                }
              },
              "application/json": {
                "schema": {
                  "type": "array",
                  "items": {
                    "$ref": "#/components/schemas/Map"
                  }
                }
              },
              "text/json": {
                "schema": {
                  "type": "array",
                  "items": {
                    "$ref": "#/components/schemas/Map"
                  }
                }
              }
            }
          }
        }
      },
      "post": {
        "tags": [
          "Maps"
        ],
        "summary": "Creates a map.",
        "description": "Sample request:\r\n\r\n    POST /api/maps\r\n    {\r\n      \"Columns\": 5,\r\n      \"Rows\": 6,\r\n      \"Name\": \"Map 5\",\r\n      \"Description\": \"Not Square\"\r\n    }",
        "requestBody": {
          "description": "A new map from the HTTP request.",
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/Map"
              }
            },
            "text/json": {
              "schema": {

```

Pretty-print

```

    },
    "application/*+json": {
      "schema": {
        "$ref": "#/components/schemas/Map"
      }
    }
  },
  "responses": {
    "201": {
      "description": "Returns the newly created map"
    },
    "400": {
      "description": "If the map is null, or addition fails",
      "content": {
        "text/plain": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        },
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      }
    },
    "409": {
      "description": "If a map with the same name already exists.",
      "content": {
        "text/plain": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        },
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      }
    }
  }
},
"/api/maps/square": {
  "get": {
    "tags": [
      "Maps"
    ],
    "summary": "Retrieves only square maps.",
    "responses": {
      "200": {
        "description": "Returns Ok with the list of maps that are square",
        "content": {
          "text/plain": {
            "schema": {

```

Pretty-print

```

        "items": {
          "$ref": "#/components/schemas/Map"
        }
      },
      "application/json": {
        "schema": {
          "type": "array",
          "items": {
            "$ref": "#/components/schemas/Map"
          }
        }
      },
      "text/json": {
        "schema": {
          "type": "array",
          "items": {
            "$ref": "#/components/schemas/Map"
          }
        }
      }
    }
  },
  "/api/maps/{id}": {
    "get": {
      "tags": [
        "Maps"
      ],
      "summary": "Retrieves a map by its ID.",
      "operationId": "GetMap",
      "parameters": [
        {
          "name": "id",
          "in": "path",
          "description": "The ID of the map to retrieve.",
          "required": true,
          "schema": {
            "type": "integer",
            "format": "int32"
          }
        }
      ],
      "responses": {
        "404": {
          "description": "If the Map was not found",
          "content": {
            "text/plain": {
              "schema": {
                "$ref": "#/components/schemas/ProblemDetails"
              }
            },
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/ProblemDetails"
              }
            },
            "text/json": {
              "schema": {
                "$ref": "#/components/schemas/ProblemDetails"
              }
            }
          }
        },
        "200": {
          "description": "Returns Ok with the map object"
        }
      }
    }
  }
}

```

Pretty-print

```

    },
    "put": {
      "tags": [
        "Maps"
      ],
      "summary": "Updates an existing map.",
      "description": "Sample request:\r\n      \r\n      PUT /api/maps/:id\r\n      {\r\n        \"Columns\": 5,\r\n        \"Rows\": 5,\r\n        \"Name\": \"Map 4\"\r\n      }",
      "parameters": [
        {
          "name": "id",
          "in": "path",
          "description": "The ID of the map to update.",
          "required": true,
          "schema": {
            "type": "integer",
            "format": "int32"
          }
        }
      ],
      "requestBody": {
        "description": "The updated map details from the HTTP request.",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/Map"
            }
          },
          "text/json": {
            "schema": {
              "$ref": "#/components/schemas/Map"
            }
          },
          "application/*+json": {
            "schema": {
              "$ref": "#/components/schemas/Map"
            }
          }
        }
      },
      "responses": {
        "404": {
          "description": "If the map isn't found in database",
          "content": {
            "text/plain": {
              "schema": {
                "$ref": "#/components/schemas/ProblemDetails"
              }
            },
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/ProblemDetails"
              }
            },
            "text/json": {
              "schema": {
                "$ref": "#/components/schemas/ProblemDetails"
              }
            }
          }
        },
        "204": {
          "description": "No Content if map is updated sucessfully"
        },
        "400": {
          "description": "If update failed.",
          "content": {

```

Pretty-print

```

    "schema": {
      "$ref": "#/components/schemas/ProblemDetails"
    },
    "application/json": {
      "schema": {
        "$ref": "#/components/schemas/ProblemDetails"
      }
    },
    "text/json": {
      "schema": {
        "$ref": "#/components/schemas/ProblemDetails"
      }
    }
  }
},
"delete": {
  "tags": [
    "Maps"
  ],
  "summary": "Deletes an existing map.",
  "parameters": [
    {
      "name": "id",
      "in": "path",
      "description": "The ID of the map to delete.",
      "required": true,
      "schema": {
        "type": "integer",
        "format": "int32"
      }
    }
  ],
  "responses": {
    "404": {
      "description": "If the map isn't found in database",
      "content": {
        "text/plain": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        },
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      }
    },
    "204": {
      "description": "No Content if Map was deleted"
    }
  }
},
"/api/maps/{id}/{x}-{y}": {
  "get": {
    "tags": [
      "Maps"
    ],
    "summary": "// Check whether the coordinate fits in a specific map.",

```

Pretty-print

```
{
  "name": "id",
  "in": "path",
  "description": "The ID of the map that the coordinate is checking it fits in.",
  "required": true,
  "schema": {
    "type": "integer",
    "format": "int32"
  }
},
{
  "name": "x",
  "in": "path",
  "description": "The x value of the coordinate.",
  "required": true,
  "schema": {
    "type": "integer",
    "format": "int32"
  }
},
{
  "name": "y",
  "in": "path",
  "description": "The y value of the coordinate.",
  "required": true,
  "schema": {
    "type": "integer",
    "format": "int32"
  }
}
],
"responses": {
  "400": {
    "description": "If the coordinate values weren't negative",
    "content": {
      "text/plain": {
        "schema": {
          "$ref": "#/components/schemas/ProblemDetails"
        }
      },
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/ProblemDetails"
        }
      },
      "text/json": {
        "schema": {
          "$ref": "#/components/schemas/ProblemDetails"
        }
      }
    }
  },
  "404": {
    "description": "A map that matches in the input id wasn't found",
    "content": {
      "text/plain": {
        "schema": {
          "$ref": "#/components/schemas/ProblemDetails"
        }
      },
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/ProblemDetails"
        }
      },
      "text/json": {
        "schema": {
          "$ref": "#/components/schemas/ProblemDetails"
        }
      }
    }
  }
}
```

Pretty-print

```

    }
  },
  "200": {
    "description": "Returns Ok with the boolean result of whether the coordinate is on the
map"
  }
}
},
"/api/robot-commands": {
  "get": {
    "tags": [
      "RobotCommands"
    ],
    "summary": "Retrieves all robot commands.",
    "responses": {
      "200": {
        "description": "Returns Ok with the list of robot commands",
        "content": {
          "text/plain": {
            "schema": {
              "type": "array",
              "items": {
                "$ref": "#/components/schemas/RobotCommand"
              }
            }
          },
          "application/json": {
            "schema": {
              "type": "array",
              "items": {
                "$ref": "#/components/schemas/RobotCommand"
              }
            }
          },
          "text/json": {
            "schema": {
              "type": "array",
              "items": {
                "$ref": "#/components/schemas/RobotCommand"
              }
            }
          }
        }
      }
    }
  },
  "post": {
    "tags": [
      "RobotCommands"
    ],
    "summary": "Creates a robot command.",
    "description": "Sample request:\r\n          \r\n          POST /api/robot-commands\r\n          {\r\n            \"Name\": \"DOWN\", \r\n            \"IsMoveCommand\": false\r\n          }",
    "requestBody": {
      "description": "A new robot command from the HTTP request.",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/RobotCommand"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/RobotCommand"
          }
        }
      }
    }
  },

```

Pretty-print

```

    "schema": {
      "$ref": "#/components/schemas/RobotCommand"
    }
  },
  "responses": {
    "201": {
      "description": "Returns the newly created robot command"
    },
    "400": {
      "description": "If the robot command is null, or addition fails",
      "content": {
        "text/plain": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        },
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      }
    },
    "409": {
      "description": "If a robot command with the same name already exists.",
      "content": {
        "text/plain": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        },
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      }
    }
  }
},
"/api/robot-commands/move": {
  "get": {
    "tags": [
      "RobotCommands"
    ],
    "summary": "Retrieves only robot commands that move the robot.",
    "responses": {
      "200": {
        "description": "Returns Ok with the list of robot commands that move",
        "content": {
          "text/plain": {
            "schema": {
              "type": "array",
              "items": {
                "$ref": "#/components/schemas/RobotCommand"
              }
            }
          }
        }
      }
    }
  }
}

```


Pretty-print

```

    },
    "application/json": {
      "schema": {
        "type": "array",
        "items": {
          "$ref": "#/components/schemas/RobotCommand"
        }
      }
    },
    "text/json": {
      "schema": {
        "type": "array",
        "items": {
          "$ref": "#/components/schemas/RobotCommand"
        }
      }
    }
  }
},
"/api/robot-commands/{id}": {
  "get": {
    "tags": [
      "RobotCommands"
    ],
    "summary": "Retrieves a robot command by its ID.",
    "operationId": "GetRobotCommand",
    "parameters": [
      {
        "name": "id",
        "in": "path",
        "description": "The ID of the robot command to retrieve.",
        "required": true,
        "schema": {
          "type": "integer",
          "format": "int32"
        }
      }
    ],
    "responses": {
      "404": {
        "description": "If the RobotCommand was not found",
        "content": {
          "text/plain": {
            "schema": {
              "$ref": "#/components/schemas/ProblemDetails"
            }
          }
        },
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      },
      "200": {
        "description": "Returns Ok with the command object"
      }
    }
  }
},

```

Pretty-print

```

tags: [
  "RobotCommands"
],
"summary": "Updates an existing robot command.",
"description": "Sample request:\r\n      \r\n      PUT /api/robot-commands/:id\r\n{\r\n  \"Name\": \"DIAGONAL\", \r\n  \"Description\": \"Description of the new\r\ncommand\", \r\n  \"IsMoveCommand\": false\r\n}",
"parameters": [
  {
    "name": "id",
    "in": "path",
    "description": "The ID of the command to update.",
    "required": true,
    "schema": {
      "type": "integer",
      "format": "int32"
    }
  }
],
"requestBody": {
  "description": "The updated command details from the HTTP request.",
  "content": {
    "application/json": {
      "schema": {
        "$ref": "#/components/schemas/RobotCommand"
      }
    },
    "text/json": {
      "schema": {
        "$ref": "#/components/schemas/RobotCommand"
      }
    },
    "application/*+json": {
      "schema": {
        "$ref": "#/components/schemas/RobotCommand"
      }
    }
  }
},
"responses": {
  "404": {
    "description": "If the robot command isn't found in database",
    "content": {
      "text/plain": {
        "schema": {
          "$ref": "#/components/schemas/ProblemDetails"
        }
      },
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/ProblemDetails"
        }
      },
      "text/json": {
        "schema": {
          "$ref": "#/components/schemas/ProblemDetails"
        }
      }
    }
  },
  "204": {
    "description": "No Content if Robot is updated sucessfully"
  },
  "400": {
    "description": "If update failed.",
    "content": {
      "text/plain": {
        "schema": {

```

Pretty-print

```

    },
    "application/json": {
      "schema": {
        "$ref": "#/components/schemas/ProblemDetails"
      }
    },
    "text/json": {
      "schema": {
        "$ref": "#/components/schemas/ProblemDetails"
      }
    }
  }
},
"delete": {
  "tags": [
    "RobotCommands"
  ],
  "summary": "Deletes an existing robot command.",
  "parameters": [
    {
      "name": "id",
      "in": "path",
      "description": "The ID of the command to delete.",
      "required": true,
      "schema": {
        "type": "integer",
        "format": "int32"
      }
    }
  ],
  "responses": {
    "404": {
      "description": "If the robot command isn't found in database",
      "content": {
        "text/plain": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        },
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/ProblemDetails"
          }
        }
      }
    },
    "204": {
      "description": "No Content if Robot was deleted"
    }
  }
},
"components": {
  "schemas": {
    "Map": {
      "type": "object",
      "properties": {
        "id": {
          "type": "integer",

```

Pretty-print

```
    },
    "columns": {
      "type": "integer",
      "format": "int32"
    },
    "rows": {
      "type": "integer",
      "format": "int32"
    },
    "name": {
      "type": "string",
      "nullable": true
    },
    "description": {
      "type": "string",
      "nullable": true
    },
    "createdDate": {
      "type": "string",
      "format": "date-time"
    },
    "modifiedDate": {
      "type": "string",
      "format": "date-time"
    }
  },
  "additionalProperties": false
},
"ProblemDetails": {
  "type": "object",
  "properties": {
    "type": {
      "type": "string",
      "nullable": true
    },
    "title": {
      "type": "string",
      "nullable": true
    },
    "status": {
      "type": "integer",
      "format": "int32",
      "nullable": true
    },
    "detail": {
      "type": "string",
      "nullable": true
    },
    "instance": {
      "type": "string",
      "nullable": true
    }
  },
  "additionalProperties": {
  }
},
"RobotCommand": {
  "type": "object",
  "properties": {
    "id": {
      "type": "integer",
      "format": "int32"
    },
    "name": {
      "type": "string",
      "nullable": true
    }
  },
}
```

Pretty-print

```
    "type": "string",
    "nullable": true
  },
  "isMoveCommand": {
    "type": "boolean"
  },
  "createdDate": {
    "type": "string",
    "format": "date-time"
  },
  "modifiedDate": {
    "type": "string",
    "format": "date-time"
  }
},
"additionalProperties": false
}
}
}
```