

API Reference

OpenAPI definition

API Version: v0

INDEX

1. CROWD-PRESSURE-CONTROLLER	3
1.1 POST /api/v1/simulation/	3
1.2 PATCH /api/v1/simulation/{simulationId}/step	6
1.3 PATCH /api/v1/simulation/{simulationId}/reset	8
1.4 GET /api/v1/simulation/{simulationId}	9
1.5 DELETE /api/v1/simulation/{simulationId}	10

API

1. CROWD-PRESSURE-CONTROLLER

1.1 POST /api/v1/simulation/

Creates a new simulation and returns its state.

Creates a new simulation and returns its state. The operation may fail if the request contains invalid parameters.

REQUEST

REQUEST BODY - application/json

```
{
  roads [{
    Array of object:
    start {
      x number
      y number
    }
    end {
      x number
      y number
    }
    width number
    crossing {
      start {
        x number
        y number
      }
      end {
        x number
        y number
      }
      width number
    }
  }]
  agentGroups [{
    Array of object:
    startCenter {
      x number
      y number
    }
    startRadius number
    destination {
      x number
      y number
    }
    destinationRadius number
    groupSize integer
  }]
  simulationWidth integer
  simulationHeight integer
}
```

```

    scaleCoefficient    number
    destinationRadius    number
    timeQuantum          number
}

```

RESPONSE

STATUS CODE - 200: OK response with the current state of the simulation if it was created.

RESPONSE MODEL - */*

```

{
  response {
    body {
      simulationId      string
      board {
        width  integer
        height integer
        walls [{
          Array of object:
            start {
              x number
              y number
            }
            end {
              x number
              y number
            }
          }
        ]
      }
      agents [{
        Array of object:
          position {
            x number
            y number
          }
          agentDesiredPosition {
            x number
            y number
          }
          agentMass          number
          agentRadius         number
          agentVisionAngle   number
          agentMaxVisionDistance number
          isStopped          boolean
        }
      ]
    }
  }
  errors {
    message string
    code    enum    ALLOWED:CRITICAL_ERROR, UNKNOWN_ERROR, SIMULATION_ERROR
  }
}

```

STATUS CODE - 404: BadRequest response if any of the parameter is invalid. Each parameter has to be non-negative and there has to be at least one road. Each road has to have exactly one crossing defined. There has to be at least one agent group.

RESPONSE MODEL - */*

```

{
  response {
    body {
      simulationId      string
      board {
        width  integer
        height integer
        walls [{
          Array of object:
            start {
              x number
              y number
            }
            end {
              x number
              y number
            }
          }
        }
      }
      agents [{
        Array of object:
          position {
            x number
            y number
          }
          agentDesiredPosition {
            x number
            y number
          }
          agentMass          number
          agentRadius         number
          agentVisionAngle   number
          agentMaxVisionDistance number
          isStopped          boolean
        }
      ]
    }
  }
  errors {
    message string
    code    enum    ALLOWED:CRITICAL_ERROR, UNKNOWN_ERROR, SIMULATION_ERROR
  }
}

```

STATUS CODE - 500: ServerSide error if the simulation couldn't be initialized.

RESPONSE MODEL - */*

```

{
  response {
    body {
      simulationId      string
      board {
        width  integer
        height integer
        walls [{
          Array of object:
            start {
              x number
              y number
            }
          }
        }
      }
    }
  }
}

```

```

    }
    end {
      x number
      y number
    }
  }
}]
}
agents [{
  Array of object:
  position {
    x number
    y number
  }
  agentDesiredPosition {
    x number
    y number
  }
  agentMass number
  agentRadius number
  agentVisionAngle number
  agentMaxVisionDistance number
  isStopped boolean
}]
}
}
errors {
  message string
  code enum    ALLOWED:CRITICAL_ERROR, UNKNOWN_ERROR, SIMULATION_ERROR
}
}

```

1.2 PATCH /api/v1/simulation/{simulationId}/step

Executes given amount of steps in a given simulation.

Executes given amount of steps in a given simulation and returns its latest state. Doesn't make any effect if the simulation doesn't exist. The operation is synchronous and may take significant amount of time.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*simulationId	string	The id of the simulation to execute steps.

QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
*steps	int32	The amount of steps to execute.

RESPONSE

STATUS CODE - 200: OK response with the latest state of the simulation if it exist.

RESPONSE MODEL - */*

```
{
  response {
    body {
      simulationId      string
      board {
        width  integer
        height integer
        walls [{
          Array of object:
            start {
              x number
              y number
            }
            end {
              x number
              y number
            }
          }
        ]
      }
      agents [{
        Array of object:
          position {
            x number
            y number
          }
          agentDesiredPosition {
            x number
            y number
          }
          agentMass          number
          agentRadius         number
          agentVisionAngle   number
          agentMaxVisionDistance number
          isStopped          boolean
        }
      ]
    }
  }
  errors {
    message string
    code    enum    ALLOWED:CRITICAL_ERROR, UNKNOWN_ERROR, SIMULATION_ERROR
  }
}
```

STATUS CODE - 500: ServerSide error if the simulation cannot execute the given amount of steps.

RESPONSE MODEL - */*

```
{
  response {
    body {
      simulationId      string
      board {
        width  integer
        height integer
        walls [{
          Array of object:
            start {
```

```

        x number
        y number
    }
    end {
        x number
        y number
    }
}]]
}
agents [{
    Array of object:
    position {
        x number
        y number
    }
    agentDesiredPosition {
        x number
        y number
    }
    agentMass                number
    agentRadius               number
    agentVisionAngle          number
    agentMaxVisionDistance    number
    isStopped                  boolean
}]
}
errors {
    message string
    code      enum    ALLOWED:CRITICAL_ERROR, UNKNOWN_ERROR, SIMULATION_ERROR
}
}

```

1.3 PATCH /api/v1/simulation/{simulationId}/reset

Restarts the simulation to its beginning state.

Restarts the simulation to its beginning state if the simulation exists.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*simulationId	string	The id of the simulation to restart.

RESPONSE

STATUS CODE - 200: OK response with the restarted state of the simulation if it existed.

RESPONSE MODEL - */*

```

{
  response {
    body {
      simulationId    string
    }
  }
}

```



```

board {
  width  integer
  height integer
  walls [{
    Array of object:
    start {
      x number
      y number
    }
    end {
      x number
      y number
    }
  }]
}
agents [{
  Array of object:
  position {
    x number
    y number
  }
  agentDesiredPosition {
    x number
    y number
  }
  agentMass          number
  agentRadius         number
  agentVisionAngle    number
  agentMaxVisionDistance number
  isStopped           boolean
}]
}
errors {
  message string
  code    enum    ALLOWED:CRITICAL_ERROR, UNKNOWN_ERROR, SIMULATION_ERROR
}
}

```

1.4 GET /api/v1/simulation/{simulationId}

Returns the current state of the simulation with given simulation id.

Returns the current state of the simulation with given simulation id if the simulation exists. Otherwise returns null.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*simulationId	string	The id of the simulation to return.

RESPONSE

STATUS CODE - 200: OK response with the current state of the simulation if it exists.

RESPONSE MODEL - */*

```
{
  response {
    body {
      simulationId      string
      board {
        width  integer
        height integer
        walls [{
          Array of object:
            start {
              x number
              y number
            }
            end {
              x number
              y number
            }
          }
        ]
      }
      agents [{
        Array of object:
          position {
            x number
            y number
          }
          agentDesiredPosition {
            x number
            y number
          }
          agentMass          number
          agentRadius         number
          agentVisionAngle   number
          agentMaxVisionDistance number
          isStopped          boolean
        }
      ]
    }
  }
  errors {
    message string
    code    enum    ALLOWED:CRITICAL_ERROR, UNKNOWN_ERROR, SIMULATION_ERROR
  }
}
```

1.5 DELETE /api/v1/simulation/{simulationId}

Deletes the simulation by the id and return its last state.

Deletes the simulation by the id and return its last state. Doesn't have any effect when the simulation doesn't exist.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*simulationId	string	The id of the simulation to delete.

RESPONSE

STATUS CODE - 200: OK response with the last state of the simulation if it existed.

RESPONSE MODEL - */*

```
{
  response {
    body {
      simulationId      string
      board {
        width  integer
        height integer
        walls [{
          Array of object:
            start {
              x number
              y number
            }
            end {
              x number
              y number
            }
          }]
      }
      agents [{
        Array of object:
          position {
            x number
            y number
          }
          agentDesiredPosition {
            x number
            y number
          }
          agentMass          number
          agentRadius         number
          agentVisionAngle   number
          agentMaxVisionDistance number
          isStopped           boolean
        }]
    }
  }
  errors {
    message string
    code    enum    ALLOWED:CRITICAL_ERROR, UNKNOWN_ERROR, SIMULATION_ERROR
  }
}
```