

# Protocolspecificatie TIN COS01

Nitin Lala: 1051727

Kadirhan Akin:

David Akerboom:

Tim Kieboom:

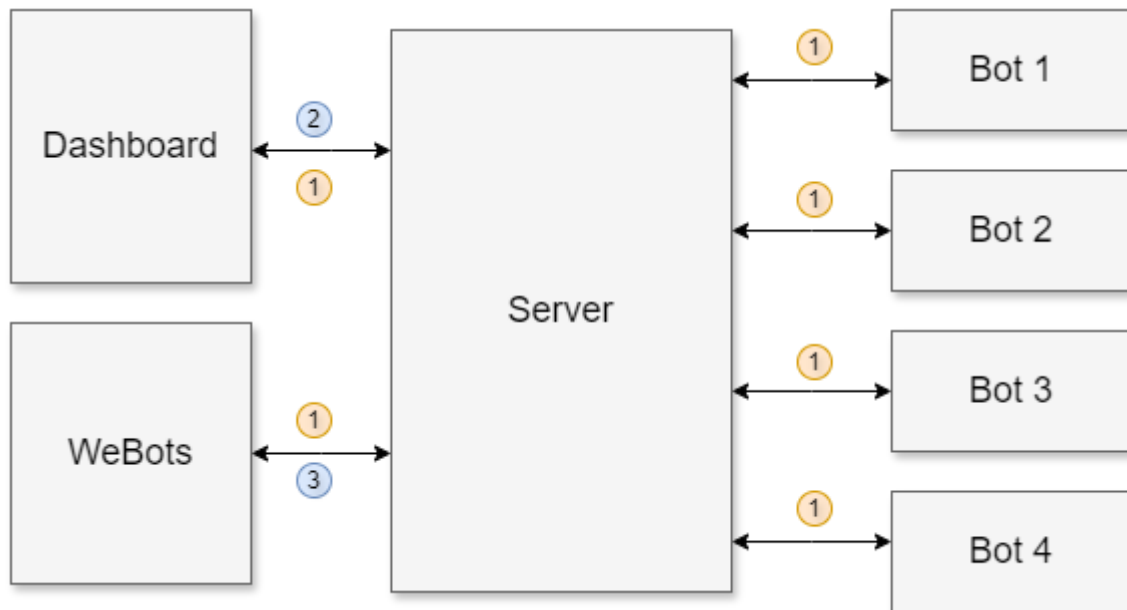
## Inhoud

1. Overzicht .....	2
2. Berichten .....	3
2.1 Protocol 1 .....	3
2.1.1 GetBotPositions.....	3
2.2 Protocol 2 .....	4
2.2.1 AddToQueue .....	4
2.2.2 SetTarget .....	5
2.2.3 GetFirst5Queues .....	6
2.2.4 ToggleMoveBots.....	6
2.3 Protocol 3 .....	8
2.3.1 NextPosition .....	8
3. Bijlage: OpenAPI Specificatie .....	9

## 1. Overzicht

Dit document bevat de specificatie van het communicatie protocol dat wij hebben gehanteerd voor de opdracht in het kader van het practicum Connected Systems (TIN COS01).

Hieronder is het gebouwde systeem afgebeeld met de verschillende onderdelen:



Bot 1 t/m Bot 4 zijn losse units (ESP32) die middels het HTTP protocol een GET doen bij de server om de huidige toestand op te halen. Het oranje bolletje met cijfer "1" staat voor dit verkeer.

De andere partijen (het dashboard en de WeBots omgeving) gebruiken ook hetzelfde bericht, dus bij deze onderdelen is ook een bolletje met het cijfer "1" te zien.

Het dashboard heeft additionele berichten die hier afgebeeld zijn als een blauw bolletje met het cijfer "2" erin. Deze communicatie is kortweg: (overigens worden alle berichten uitgebreid besproken in het volgende hoofdstuk).

- De target zetten voor een bot.
- De simulatie stoppen en starten.
- De eerste queue informatie ophalen.

De WeBots omgeving heeft ook een additioneel bericht, dat hier afgebeeld wordt als een blauw bolletje met het getal "3". Dit bericht kan commando's vanuit WeBots sturen om instructies aan de server te geven (zie de documentatie in het volgende hoofdstuk).

In de bijlage is de OpenAPI specificatie van het protocol opgenomen. Deze specificatie beschrijft exact de verschillende berichten inclusief het JSON Schema met de mogelijke waarden, types en URL's.

## 2. Berichten

### 2.1 Protocol 1

Dit protocol wordt gebruikt door alle partijen. Het werd in het overzicht weergegevens als een oranje bolletje met het cijfer “1”.

#### 2.1.1 GetBotPositions

##### GET /api/getBotPositions

Met dit bericht vraagt een partij om de posities van de bots. De response is een array waarin voor elke bot de volgende gegevens zijn opgenomen:

- Het ID van de bot
- De richting van de bot (dit is een getal waarbij 0 rechts betekent, 1 omhoog, 2 links, 3 omlaag en 4 idle.
- De huidige positie van de bot.

##### Een voorbeeld response body

```
{
  "botPositions":
  [
    {
      "botID": 1,
      "direction": 1, // right = 0, up = 1, left = 2, down = 3, idle = 4
      "currentPosition": { "X": 1, "Y": 1 }
    }
  ]
}
```

##### De HTTP statuscodes

200 (OK): de operatie is succesvol en de response body is gevuld.

404 (NotFound): de URL is niet correct en kan niet worden gevonden.

400 (BadRequest): de request body is niet correct geformateerd.

500 (ServerError): de server heeft fouten en kan het bericht niet verwerken.

## 2.2 Protocol 2

Dit protocol wordt gebruikt door het dashboard. Het werd in het overzicht weergegevens als een blauw bolletje met het cijfer “2”.

### 2.2.1 AddToQueue

#### POST /api/addToQueue

Met dit bericht kan de het dashboard een route aanvragen bij de server. Deze route bestaat uit een pickup en een drop locatie. De server antwoord met een bevestiging waarin de botID staat van de bot die dit gaat uitvoeren en of het in de queue is gezet.

#### Een voorbeeld request body

```
{  
  "pickup": { "X": 1, "Y": 1 },  
  "drop": { "X": 1, "Y": 1 }  
}
```

#### Een voorbeeld response Body

```
{  
  "botID": 1,  
  "IsInQueue": true  
}
```

#### De HTTP statuscodes

200 (OK): de operatie is succesvol en de response body is gevuld.

404 (NotFound): de URL is niet correct en kan niet worden gevonden.

400 (BadRequest): de request body is niet correct geformateerd.

500 (ServerError): de server heft fouten en kan het bericht niet verwerken.

### 2.2.2 SetTarget

#### POST /api/setTarget

Met dit bericht kan een positie en een target gezet worden voor een specifieke bot. De server antwoord met een bevestiging en de route die is gepland.

#### Een voorbeeld request body

```
{  
  "botID": 1  
  "currentPosition": { "X": 1, "Y": 1 },  
  "targetPosition": { "X": 9, "Y": 2 }  
}
```

#### Een voorbeeld response body

```
{  
  "botID": 1,  
  "route": "(x:1,y:1) -> (x:1,y:2)"  
}
```

#### De HTTP statuscodes

200 (OK): de operatie is succesvol en de response body is gevuld.

404 (NotFound): de URL is niet correct en kan niet worden gevonden.

400 (BadRequest): de request body is niet correct geformatteerd.

500 (ServerError): de server heeft fouten en kan het bericht niet verwerken.

### 2.2.3 GetFirst5Queues

#### GET /api/getFirst5Queues

Het dashboard vraagt hiermee de bot queue informatie op om af te beelden.

#### Een voorbeeld response body

```
{
  "queues": [
    "(x:1,y:1 -> x:2,y:2)",
    "(x:3,y:6 -> x:4,y:8)",
    "(x:1,y:5 -> x:2,y:7)",
    "(x:1,y:8 -> x:1,y:2)"
  ]
}
```

#### De HTTP statuscodes

200 (OK): de operatie is succesvol en de response body is gevuld.

404 (NotFound): de URL is niet correct en kan niet worden gevonden.

400 (BadRequest): de request body is niet correct geformatteerd.

500 (ServerError): de server heeft fouten en kan het bericht niet verwerken.

### 2.2.4 ToggleMoveBots

#### POST /API/toggleMoveBots

Met dit bericht start of stopt het dashboard de simulatie. Een waarde van "true" start de simulatie en "false" pauseert de simulatie.

#### Een voorbeeld request body

```
{
  "mode": true
}
```

**De HTTP statuscodes**

200 (OK): de operatie is succesvol en de response body is gevuld.

404 (NotFound): de URL is niet correct en kan niet worden gevonden.

400 (BadRequest): de request body is niet correct geformatteerd.

500 (ServerError): de server heeft fouten en kan het bericht niet verwerken.

## 2.3 Protocol 3

Dit protocol wordt gebruikt door het dashboard. Het werd in het overzicht weergegevens als een blauw bolletje met het cijfer “3”.

### 2.3.1 NextPosition

#### POST /api/nextPosition

Wordt gebruikt door WeBots om informatie voor een bot te zetten.

#### Een voorbeeld request body

```
{
  "botID": 1,
  "nextPosition": { 1, 1 },
  "targetPosition": { 3, 3 },
  "direction": 1, // right = 0, up = 1, left = 2, down = 3, idle = 4
  "detectedObjects": [
    { 1, 1 },
    { 3, 5 }
  ]
}
```

#### De HTTP statuscodes

200 (OK): de operatie is succesvol en de response body is gevuld.

404 (NotFound): de URL is niet correct en kan niet worden gevonden.

400 (BadRequest): de request body is niet correct geformatteerd.

500 (ServerError): de server heft fouten en kan het bericht niet verwerken.



### 3. Bijlage: OpenAPI Specificatie

```
{
  "openapi": "3.0.1",
  "info": {
    "title": "backend",
    "version": "1.0"
  },
  "paths": {
    "/api/setTarget": {
      "post": {
        "tags": [
          ""
        ],
        "requestBody": {
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/SendTargetRequestBody"
              }
            },
            "text/json": {
              "schema": {
                "$ref": "#/components/schemas/SendTargetRequestBody"
              }
            },
            "application/*+json": {
              "schema": {
                "$ref": "#/components/schemas/SendTargetRequestBody"
              }
            }
          }
        },
        "responses": {
          "200": {
            "description": "Success"
          }
        }
      }
    },
    "/api/getBotPositions": {
      "get": {
        "tags": [
          ""
        ],
        "responses": {
          "200": {
            "description": "Success"
          }
        }
      }
    }
  }
}
```

```

},
"/api/addToQueue": {
  "post": {
    "tags": [
      ""
    ],
    "requestBody": {
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/AddToQueueResrequestBody"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/AddToQueueResrequestBody"
          }
        },
        "application/*+json": {
          "schema": {
            "$ref": "#/components/schemas/AddToQueueResrequestBody"
          }
        }
      }
    },
    "responses": {
      "200": {
        "description": "Success"
      }
    }
  }
},
"/api/nextPosition": {
  "post": {
    "tags": [
      ""
    ],
    "requestBody": {
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/NextPositionRequestBody"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/NextPositionRequestBody"
          }
        },
        "application/*+json": {
          "schema": {

```

```

        "$ref": "#/components/schemas/NextPositionRequestBody"
      }
    }
  },
  "responses": {
    "200": {
      "description": "Success"
    }
  }
},
"/api/getFirst5Queues": {
  "post": {
    "tags": [
      ""
    ],
    "responses": {
      "200": {
        "description": "Success"
      }
    }
  }
},
"/api/toggleMoveBots": {
  "post": {
    "tags": [
      ""
    ],
    "requestBody": {
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/ToggleMoveBotsRequestBody"
          }
        },
        "text/json": {
          "schema": {
            "$ref": "#/components/schemas/ToggleMoveBotsRequestBody"
          }
        },
        "application/*+json": {
          "schema": {
            "$ref": "#/components/schemas/ToggleMoveBotsRequestBody"
          }
        }
      }
    },
    "responses": {
      "200": {
        "description": "Success"
      }
    }
  }
}

```

```

    }
  }
}
},
"components": {
  "schemas": {
    "AddToQueueResquestBody": {
      "type": "object",
      "properties": {
        "pickup": {
          "$ref": "#/components/schemas/Position"
        },
        "drop": {
          "$ref": "#/components/schemas/Position"
        }
      },
    },
    "additionalProperties": false
  },
  "NextPositionRequestBody": {
    "type": "object",
    "properties": {
      "botID": {
        "type": "integer",
        "format": "int32"
      },
      "nextPosition": {
        "$ref": "#/components/schemas/Position"
      },
      "targetPosition": {
        "$ref": "#/components/schemas/Position"
      },
      "direction": {
        "type": "integer",
        "format": "int32"
      },
      "detectedObjects": {
        "type": "array",
        "items": {
          "$ref": "#/components/schemas/Position"
        },
        "nullable": true
      }
    },
    "additionalProperties": false
  },
  "Position": {
    "type": "object",
    "properties": {
      "x": {
        "type": "integer",

```

```

        "format": "int32"
      },
      "y": {
        "type": "integer",
        "format": "int32"
      }
    },
    "additionalProperties": false
  },
  "SendTargetRequestBody": {
    "type": "object",
    "properties": {
      "botID": {
        "type": "integer",
        "format": "int32"
      },
      "currentPosition": {
        "$ref": "#/components/schemas/Position"
      },
      "targetPosition": {
        "$ref": "#/components/schemas/Position"
      }
    },
    "additionalProperties": false
  },
  "ToggleMoveBotsRequestBody": {
    "type": "object",
    "properties": {
      "mode": {
        "type": "boolean"
      }
    },
    "additionalProperties": false
  }
}

```