**CP-DIR Exercise**

Each Employee must receive the following:

* Id – number
* firstName – String with length > 1
* lastName – String with length > 1
* teamName – String with length > 1
* titles – a list of titles:
  + 0 – CEO(type: manager)
  + 1 – SoftwareDeveloper(type: developer)
  + 2 – TeamLeader(type: manager)
  + 3 – TechnologyLeader(type: manager)
* directManager – Id of the direct manager (must exist in DB and have manager title)
* phone - number

Requests format:

Address “http://localhost:8089/api/v1”

1. Add Developer “/addDeveloper”

Body:

{

"id":"2",

"firstName":"Cheche",

"lastName":"Mongo",

"teamName":"First Team",

"titles":["SoftwareDeveloper","TeamLeader"],

"directManager":"1",

"phone":"1"

}

\*Titles must contain developer type title

1. Add Manager “/addManager”

Body:

{

"id":"1",

"firstName":"Gil",

"lastName":"Shwed",

"teamName":"CheckPoint",

"titles":["CEO""],

"directManager":"1",

"phone":"1"

}

\*Titles must contain manager type title

1. Remove Employee “/removeEmployee/{id}”
2. Edit Employee “/editEmployee/{id}”

Body:

{

"id":"2",

"firstName":"Cheche",

"lastName":"Mongo",

"teamName":"First Team",

"titles":["SoftwareDeveloper","TeamLeader"],

"directManager":"1",

"phone":"1"

}

\* Every field is optional, fill only the ones you want to edit

1. Reporting Employees “/reportingEmployees/{id},{order}”

order: true for ascending | false for descending

Notes:

* As Gil Shwed is the CEO, after the first initiation, the following addManager request should be sent:

{

"id":"1",

"firstName":"Gil",

"lastName":"Shwed",

"teamName":"CP",

"titles":["CEO"],

"directManager":"1",

"phone":"123456"

}

* Instead of logging, SOPs were written in some places including catch blocks to seem them in docker logs.  
  Alternatively, they can be changed with log statements or redirection of output into log files.
* In case of success, each API request will respond with employees list after change.
* In case of failure, failure response in the following form will be returned:

{

<reason>: null

}

Reason – failure reason. (Case of reason = “Failed”, means that the failure cause most likely is entering catch block).

* To run my project in case of pull from GitHub, I did the following steps:
  + Executed “mvn clean install” from cp-dir-exercise folder
  + Copied Dockerfile from cp-dir-exercise folder into target folder
  + Executed “docker-compose up -d” from cp-dir-exercise folder