# Phone book REST API application

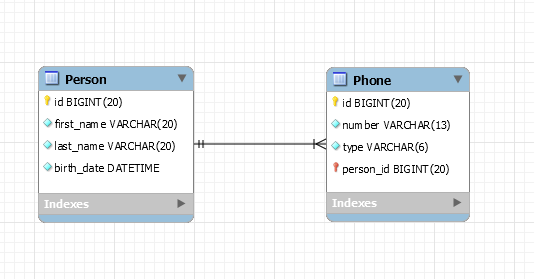
## Deliverables

* Packaged \*.jar file
* GIT repository url or packaged source code

## Requirements

* Spring boot - 2.X.X
* Packaging - Executable jar file with embedded Tomcat
* API port - 9000
* Authentication - Basic authentication
* Database - In memory database such as H2
* REST documentation - Swagger (optional)

## Data model



## Validation

Person

* id - unique identifier
* first\_name - must not be empty
* last\_name - must not be empty
* birth\_date - must not be empty and in past (format yyyy-MM-dd)

Phone

* id - unique identifier
* personId - person unique identifier
* number - must not be empty and must match "+371 XXXXXXXX"
* type - must not be empty and possible values are "mobile", "home", "work"

## Rest API methods

* GET /person
* GET /person/{id}
* POST /person
* PUT /person/{id}
* DELETE /person/{id}
* GET /phone
* GET /phone/{id}
* POST /phone
* PUT /phone/{id}
* DELETE /phone/{id}

## Custom exceptions

Throw custom exceptions if request data does not exist in database. Feel free to implement custom exceptions for other purposes.

## HTTP responses

GET

* 200 - return body
* 400 - validation exceptions
* 404 - requested data does not exist
* 500 - general exceptions

POST

* 201 - created successfully
* 400 - validation exceptions
* 404 - requested data does not exist
* 500 - general exceptions

PUT

* 200 - updated successfully
* 400 - validation exceptions
* 404 - requested data does not exist
* 500 - general exceptions

DELETE

* 200 - deleted successfully
* 404 - requested data does not exist
* 500 - general exceptions

## Exception handling and responses

Response body structure must be the same for all exceptions.

A response that have a list of exceptions (in case if there are more than one like validation exceptions).

*{*

*“exceptions”: [*

*“…….”,*

*“…….”,*

*“…….”*

*]*

*}*

## Logging

There must be a log file with Spring Boot application exceptions.