# JAM

Version 1.1.2

13.05.2025

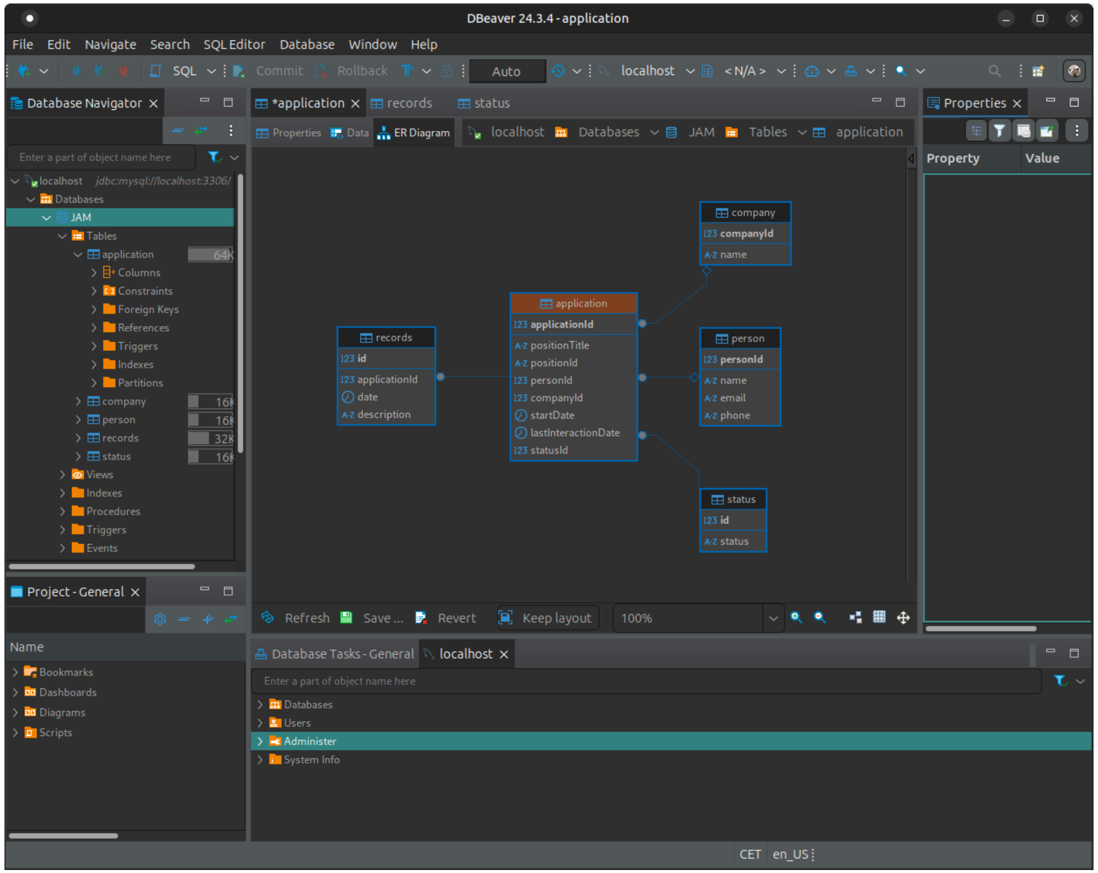
# JAM

Job Application Management – a full stack application intended to manage job applications from the perspective of an applicant.
Created using following tools and components:

# Database layer

MySQL used as base layer database.

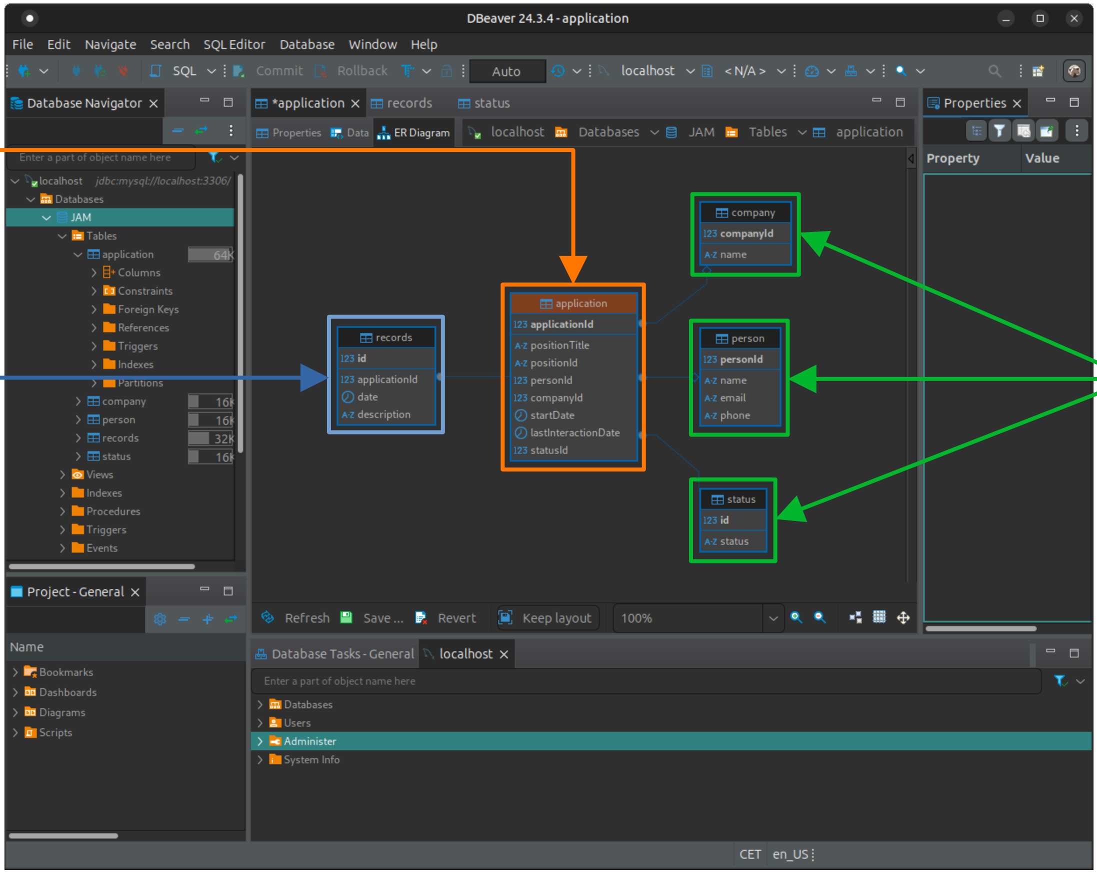
Layout of tables and views created using DBeaver



# Database layer

“application“ table used to hold information about applications

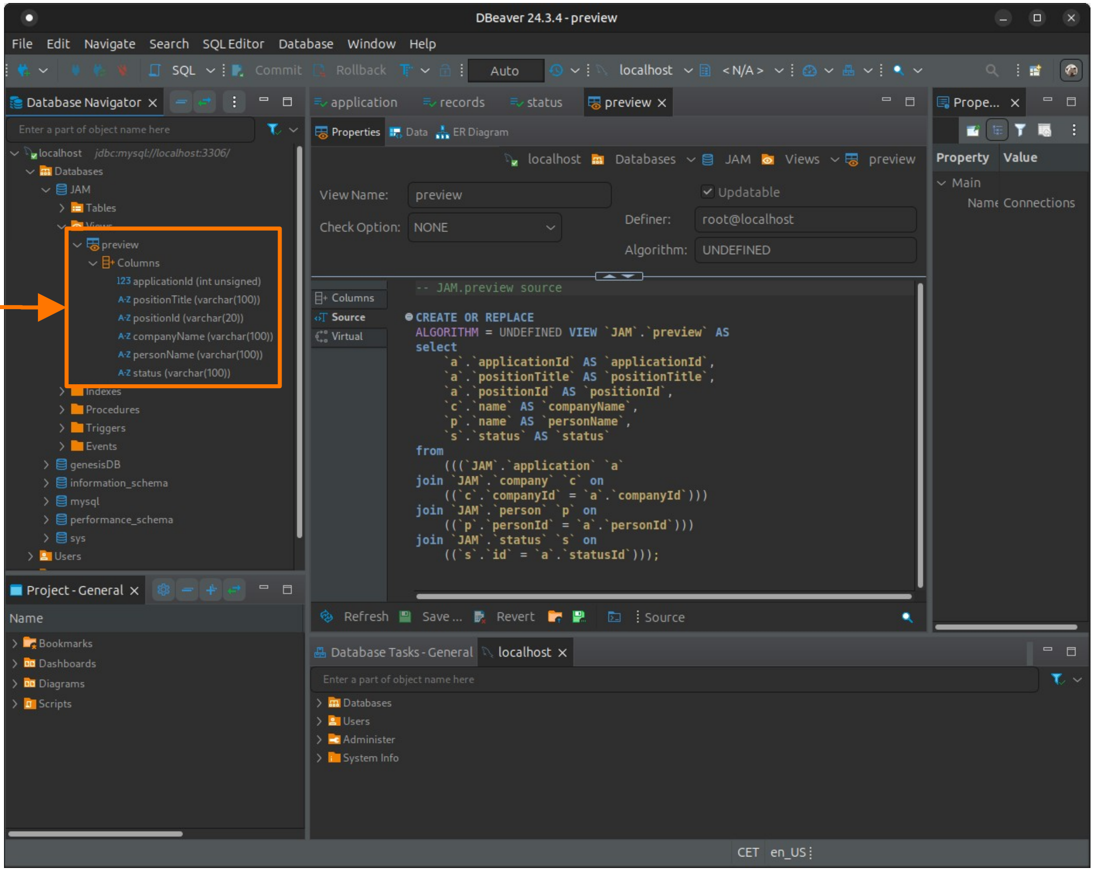
“records” table for various notes related to a particular application.



Tables holding information about persons, companies and status connected to “application” table via foreign keys

# Database layer

“preview“ view created to ease access to data used for application card (see front end section)



View Name: preview
Check Option: NONE
Updatable: ✓
Definer: root@localhost
Algorithm: UNDEFINED

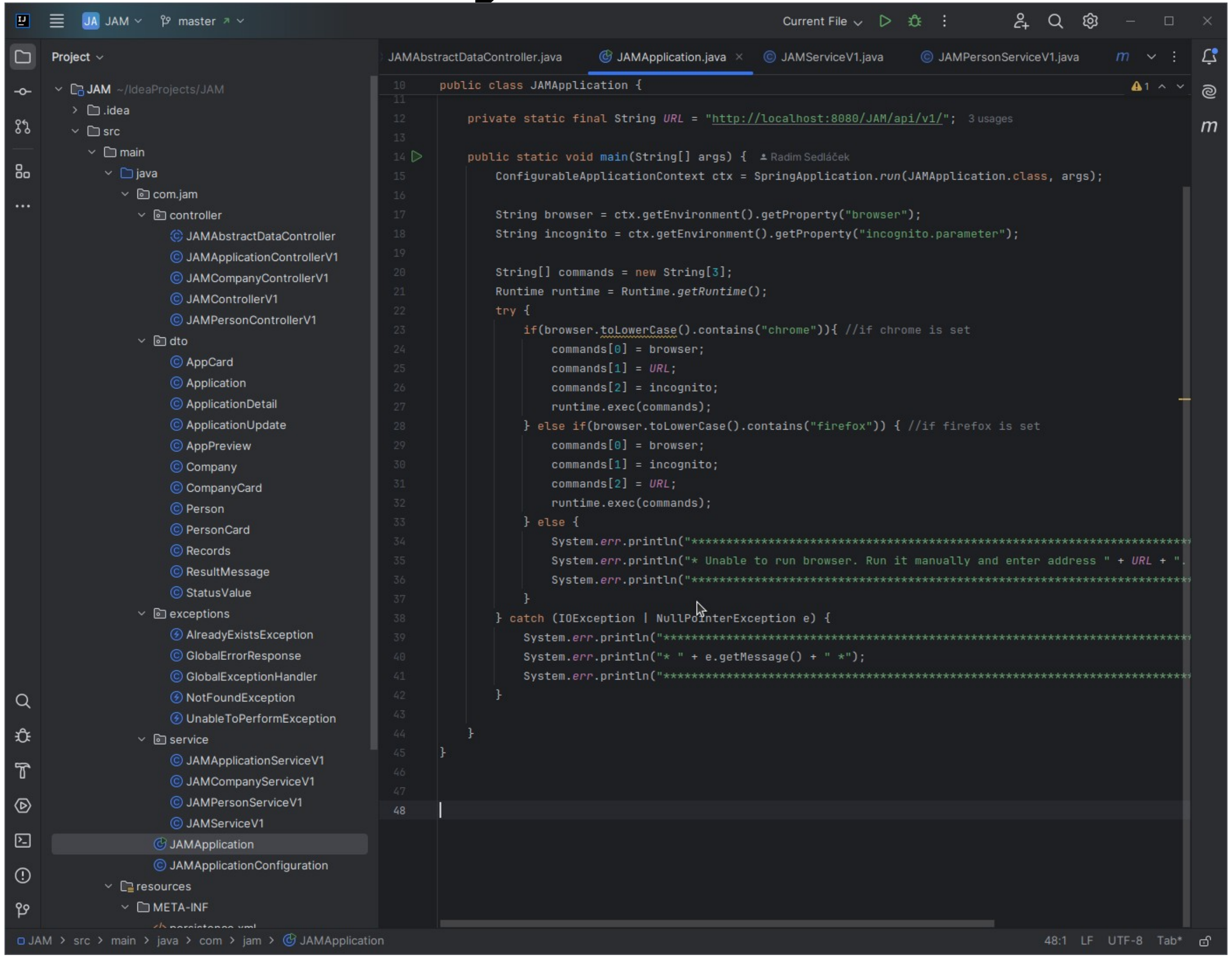
```sql
-- JAM.preview source

CREATE OR REPLACE
ALGORITHM = UNDEFINED VIEW `JAM`.`preview` AS
select
    `a`.`applicationId` AS `applicationId`,
    `a`.`positionTitle` AS `positionTitle`,
    `a`.`positionId` AS `positionId`,
    `c`.`name` AS `companyName`,
    `p`.`name` AS `personName`,
    `s`.`status` AS `status`
from
    (((`JAM`.`application` `a`
join `JAM`.`company` `c` on
    ((`c`.`companyId` = `a`.`companyId`)))
join `JAM`.`person` `p` on
    ((`p`.`personId` = `a`.`personId`)))
join `JAM`.`status` `s` on
    ((`s`.`id` = `a`.`statusId`)));
```

# Back-end layer

- Java based
- Spring boot used for REST web service/web server creation
- Maven used for packaging
- Coded in Intellij Idea IDE (community edition)
- Hibernate used for interactions with database
- Postman used for testing

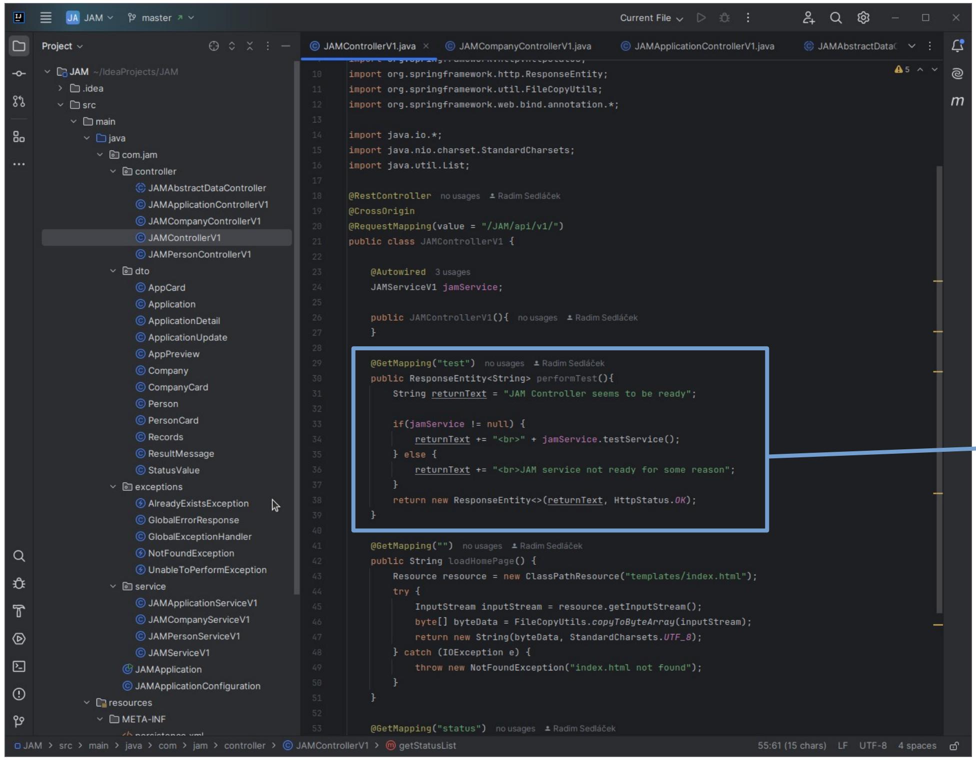
# Back-end layer

Spring boot application/controller/service leveraged for requests processing



```java
public class JAMApplication {

    private static final String URL = "http://localhost:8080/JAM/api/v1/";  3 usages

    public static void main(String[] args) {  Radim Sedláček
        ConfigurableApplicationContext ctx = SpringApplication.run(JAMApplication.class, args);

        String browser = ctx.getEnvironment().getProperty("browser");
        String incognito = ctx.getEnvironment().getProperty("incognito.parameter");

        String[] commands = new String[3];
        Runtime runtime = Runtime.getRuntime();
        try {
            if(browser.toLowerCase().contains("chrome")){ //if chrome is set
                commands[0] = browser;
                commands[1] = URL;
                commands[2] = incognito;
                runtime.exec(commands);
            } else if(browser.toLowerCase().contains("firefox")) { //if firefox is set
                commands[0] = browser;
                commands[1] = incognito;
                commands[2] = URL;
                runtime.exec(commands);
            } else {
                System.err.println("**************************************************************************
                System.err.println("* Unable to run browser. Run it manually and enter address " + URL + ".
                System.err.println("**************************************************************************
            }
        } catch (IOException | NullPointerException e) {
            System.err.println("****************************************************************************
            System.err.println("* " + e.getMessage() + " *");
            System.err.println("****************************************************************************
        }

    }
}


```

# Back-end layer

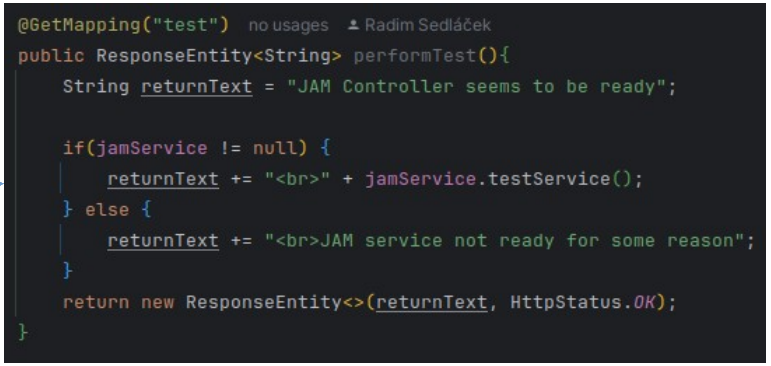


```
JAM ~/IdeaProjects/JAM
  .idea
  src
    main
      java
        com.jam
          controller
            JAMAbstractDataController
            JAMApplicationControllerV1
            JAMCompanyControllerV1
            JAMControllerV1
            JAMPersonControllerV1
          dto
            AppCard
            Application
            ApplicationDetail
            ApplicationUpdate
            AppPreview
            Company
            CompanyCard
            Person
            PersonCard
            Records
            ResultMessage
            StatusValue
          exceptions
            AlreadyExistsException
            GlobalErrorResponse
            GlobalExceptionHandler
            NotFoundException
            UnableToPerformException
          service
            JAMApplicationServiceV1
            JAMCompanyServiceV1
            JAMPersonServiceV1
            JAMServiceV1
          JAMApplication
          JAMApplicationConfiguration
      resources
        META-INF
```

```java
import org.springframework.http.ResponseEntity;
import org.springframework.util.FileCopyUtils;
import org.springframework.web.bind.annotation.*;

import java.io.*;
import java.nio.charset.StandardCharsets;
import java.util.List;

@RestController  no usages  Radim Sedláček
@CrossOrigin
@RequestMapping(value = "/JAM/api/v1/")
public class JAMControllerV1 {

    @Autowired  3 usages
    JAMServiceV1 jamService;

    public JAMControllerV1(){  no usages  Radim Sedláček
    }

    @GetMapping("test")  no usages  Radim Sedláček
    public ResponseEntity<String> performTest(){
        String returnText = "JAM Controller seems to be ready";

        if(jamService != null) {
            returnText += "<br>" + jamService.testService();
        } else {
            returnText += "<br>JAM service not ready for some reason";
        }
        return new ResponseEntity<>(returnText, HttpStatus.OK);
    }

    @GetMapping("")  no usages  Radim Sedláček
    public String loadHomePage() {
        Resource resource = new ClassPathResource("templates/index.html");
        try {
            InputStream inputStream = resource.getInputStream();
            byte[] byteData = FileCopyUtils.copyToByteArray(inputStream);
            return new String(byteData, StandardCharsets.UTF_8);
        } catch (IOException e) {
            throw new NotFoundException("index.html not found");
        }
    }

    @GetMapping("status")  no usages  Radim Sedláček
```

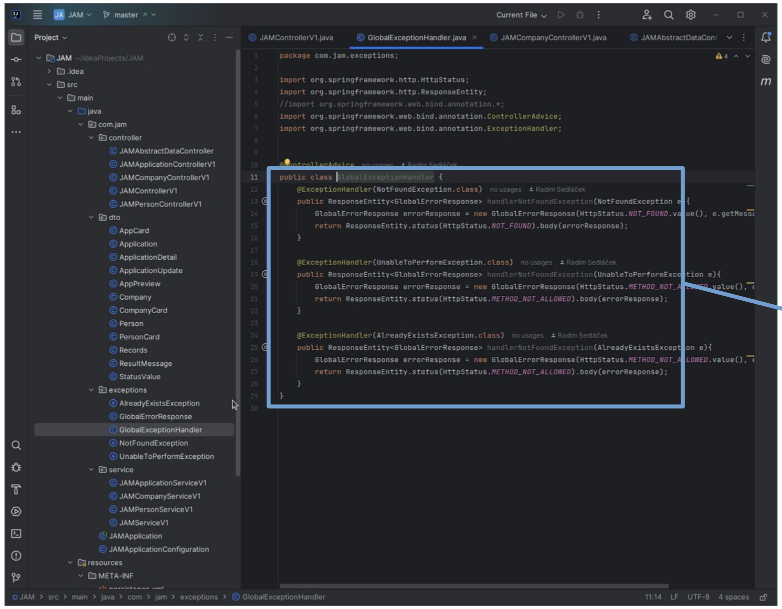
JAM
master
Current File
Project
JAMControllerV1.java
JAMCompanyControllerV1.java
JAMApplicationControllerV1.java
JAMAbstractData
JAM > src > main > java > com > jam > controller > JAMControllerV1 > getStatusList
55:61 (15 chars) LF UTF-8 4 spaces

A simple test of service and database readiness available on address “test” – see Front-end section for more details

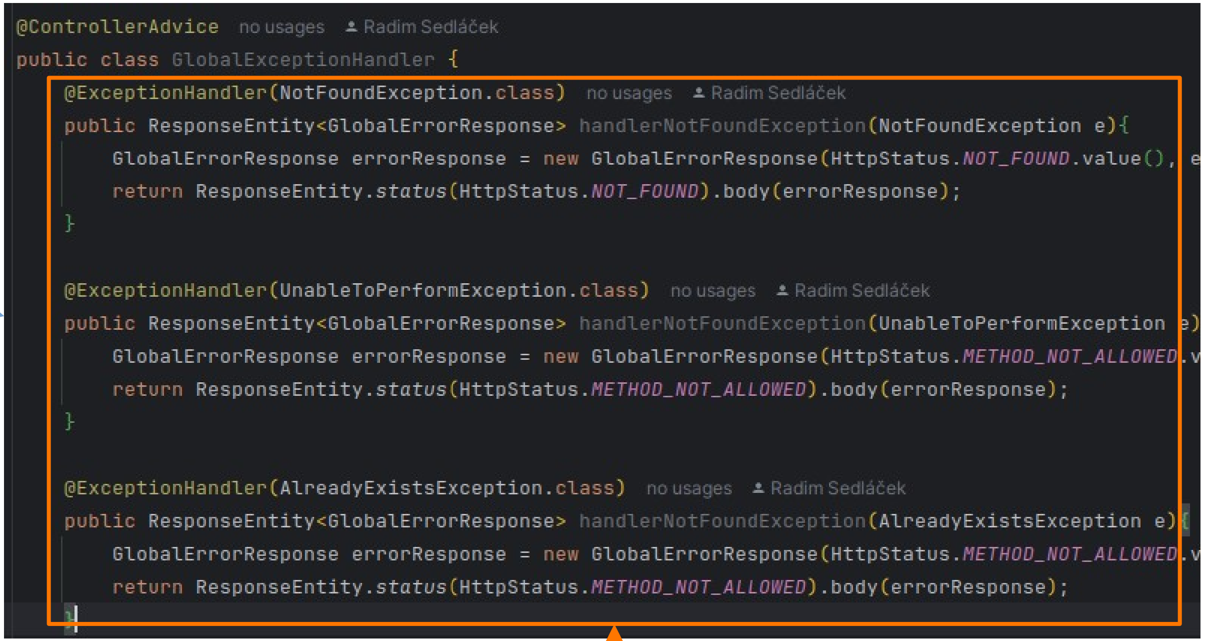


```java
@GetMapping("test")  no usages  Radim Sedláček
public ResponseEntity<String> performTest(){
    String returnText = "JAM Controller seems to be ready";

    if(jamService != null) {
        returnText += "<br>" + jamService.testService();
    } else {
        returnText += "<br>JAM service not ready for some reason";
    }
    return new ResponseEntity<>(returnText, HttpStatus.OK);
}
```

# Back-end layer



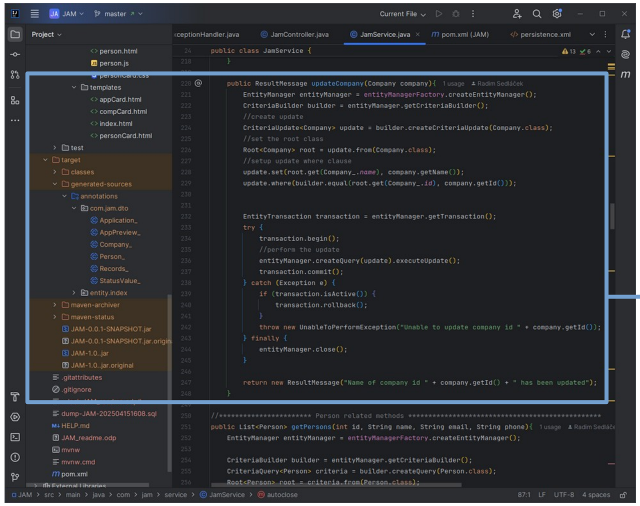
```java
package com.jam.exceptions;

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
//import org.springframework.web.bind.annotation.*;
import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;


@ControllerAdvice  no usages  Radim Sedláček
public class GlobalExceptionHandler {
    @ExceptionHandler(NotFoundException.class)  no usages  Radim Sedláček
    public ResponseEntity<GlobalErrorResponse> handlerNotFoundException(NotFoundException e){
        GlobalErrorResponse errorResponse = new GlobalErrorResponse(HttpStatus.NOT_FOUND.value(), e.getMessa
        return ResponseEntity.status(HttpStatus.NOT_FOUND).body(errorResponse);
    }

    @ExceptionHandler(UnableToPerformException.class)  no usages  Radim Sedláček
    public ResponseEntity<GlobalErrorResponse> handlerNotFoundException(UnableToPerformException e){
        GlobalErrorResponse errorResponse = new GlobalErrorResponse(HttpStatus.METHOD_NOT_ALLOWED.value(),
        return ResponseEntity.status(HttpStatus.METHOD_NOT_ALLOWED).body(errorResponse);
    }

    @ExceptionHandler(AlreadyExistsException.class)  no usages  Radim Sedláček
    public ResponseEntity<GlobalErrorResponse> handlerNotFoundException(AlreadyExistsException e){
        GlobalErrorResponse errorResponse = new GlobalErrorResponse(HttpStatus.METHOD_NOT_ALLOWED.value(),
        return ResponseEntity.status(HttpStatus.METHOD_NOT_ALLOWED).body(errorResponse);
    }
}
```



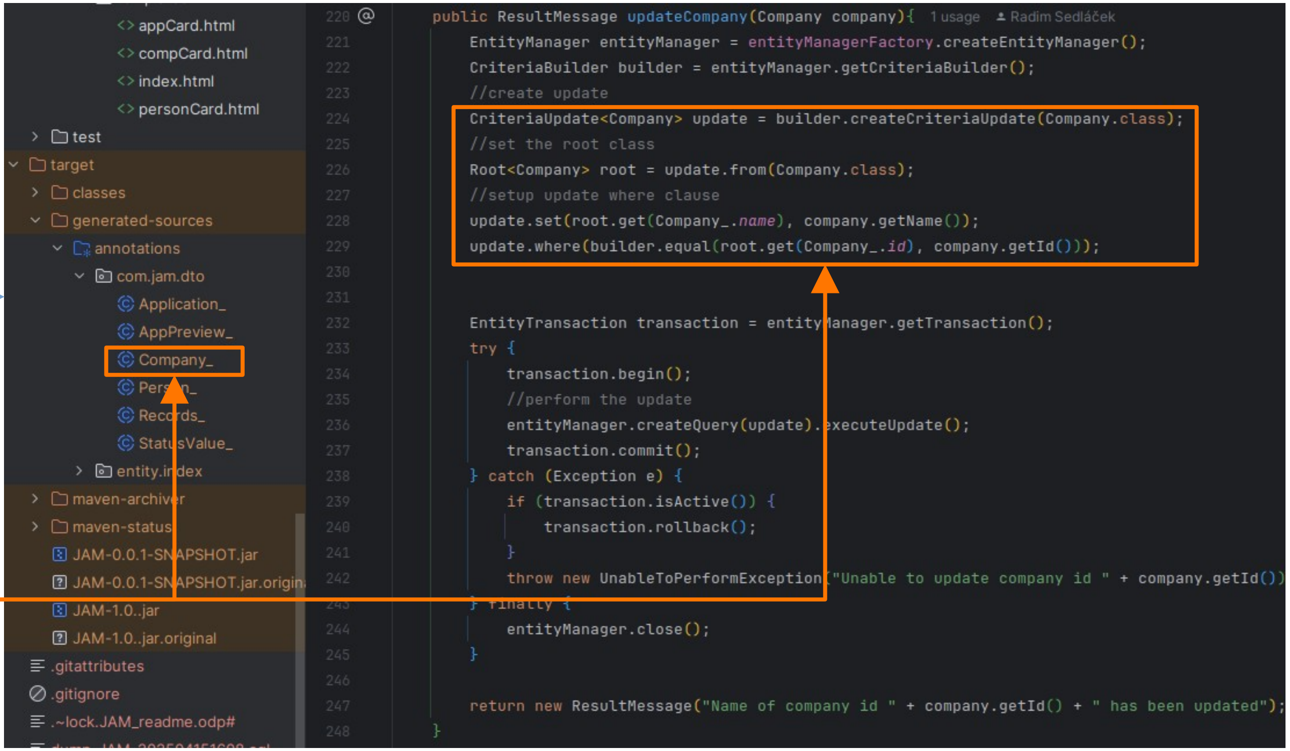
```java
@ControllerAdvice  no usages  Radim Sedláček
public class GlobalExceptionHandler {
    @ExceptionHandler(NotFoundException.class)  no usages  Radim Sedláček
    public ResponseEntity<GlobalErrorResponse> handlerNotFoundException(NotFoundException e){
        GlobalErrorResponse errorResponse = new GlobalErrorResponse(HttpStatus.NOT_FOUND.value(), e
        return ResponseEntity.status(HttpStatus.NOT_FOUND).body(errorResponse);
    }

    @ExceptionHandler(UnableToPerformException.class)  no usages  Radim Sedláček
    public ResponseEntity<GlobalErrorResponse> handlerNotFoundException(UnableToPerformException e)
        GlobalErrorResponse errorResponse = new GlobalErrorResponse(HttpStatus.METHOD_NOT_ALLOWED.v
        return ResponseEntity.status(HttpStatus.METHOD_NOT_ALLOWED).body(errorResponse);
    }

    @ExceptionHandler(AlreadyExistsException.class)  no usages  Radim Sedláček
    public ResponseEntity<GlobalErrorResponse> handlerNotFoundException(AlreadyExistsException e){
        GlobalErrorResponse errorResponse = new GlobalErrorResponse(HttpStatus.METHOD_NOT_ALLOWED.v
        return ResponseEntity.status(HttpStatus.METHOD_NOT_ALLOWED).body(errorResponse);
}
```

Spring boot @ExceptionHandler leveraged for error processing

# Back-end layer



```java
public class JamService {
    }

    public ResultMessage updateCompany(Company company){   1 usage  Radim Sedláček
        EntityManager entityManager = entityManagerFactory.createEntityManager();
        CriteriaBuilder builder = entityManager.getCriteriaBuilder();
        //create update
        CriteriaUpdate<Company> update = builder.createCriteriaUpdate(Company.class);
        //set the root class
        Root<Company> root = update.from(Company.class);
        //setup update where clause
        update.set(root.get(Company_.name), company.getName());
        update.where(builder.equal(root.get(Company_.id), company.getId()));


        EntityTransaction transaction = entityManager.getTransaction();
        try {
            transaction.begin();
            //perform the update
            entityManager.createQuery(update).executeUpdate();
            transaction.commit();
        } catch (Exception e) {
            if (transaction.isActive()) {
                transaction.rollback();
            }
            throw new UnableToPerformException("Unable to update company id " + company.getId());
        } finally {
            entityManager.close();
        }

        return new ResultMessage("Name of company id " + company.getId() + " has been updated");
    }

//*********************** Person related methods ********************************************
public List<Person> getPersons(int id, String name, String email, String phone){   1 usage  Radim Sedláče
    EntityManager entityManager = entityManagerFactory.createEntityManager();

    CriteriaBuilder builder = entityManager.getCriteriaBuilder();
    CriteriaQuery<Person> criteria = builder.createQuery(Person.class);
    Root<Person> root = criteria.from(Person.class);
```

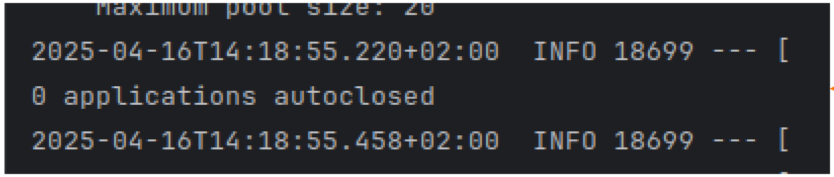


```java
public ResultMessage updateCompany(Company company){   1 usage  Radim Sedláček
    EntityManager entityManager = entityManagerFactory.createEntityManager();
    CriteriaBuilder builder = entityManager.getCriteriaBuilder();
    //create update
    CriteriaUpdate<Company> update = builder.createCriteriaUpdate(Company.class);
    //set the root class
    Root<Company> root = update.from(Company.class);
    //setup update where clause
    update.set(root.get(Company_.name), company.getName());
    update.where(builder.equal(root.get(Company_.id), company.getId()));


    EntityTransaction transaction = entityManager.getTransaction();
    try {
        transaction.begin();
        //perform the update
        entityManager.createQuery(update).executeUpdate();
        transaction.commit();
    } catch (Exception e) {
        if (transaction.isActive()) {
            transaction.rollback();
        }
        throw new UnableToPerformException("Unable to update company id " + company.getId())
    } finally {
        entityManager.close();
    }

    return new ResultMessage("Name of company id " + company.getId() + " has been updated");
}
```

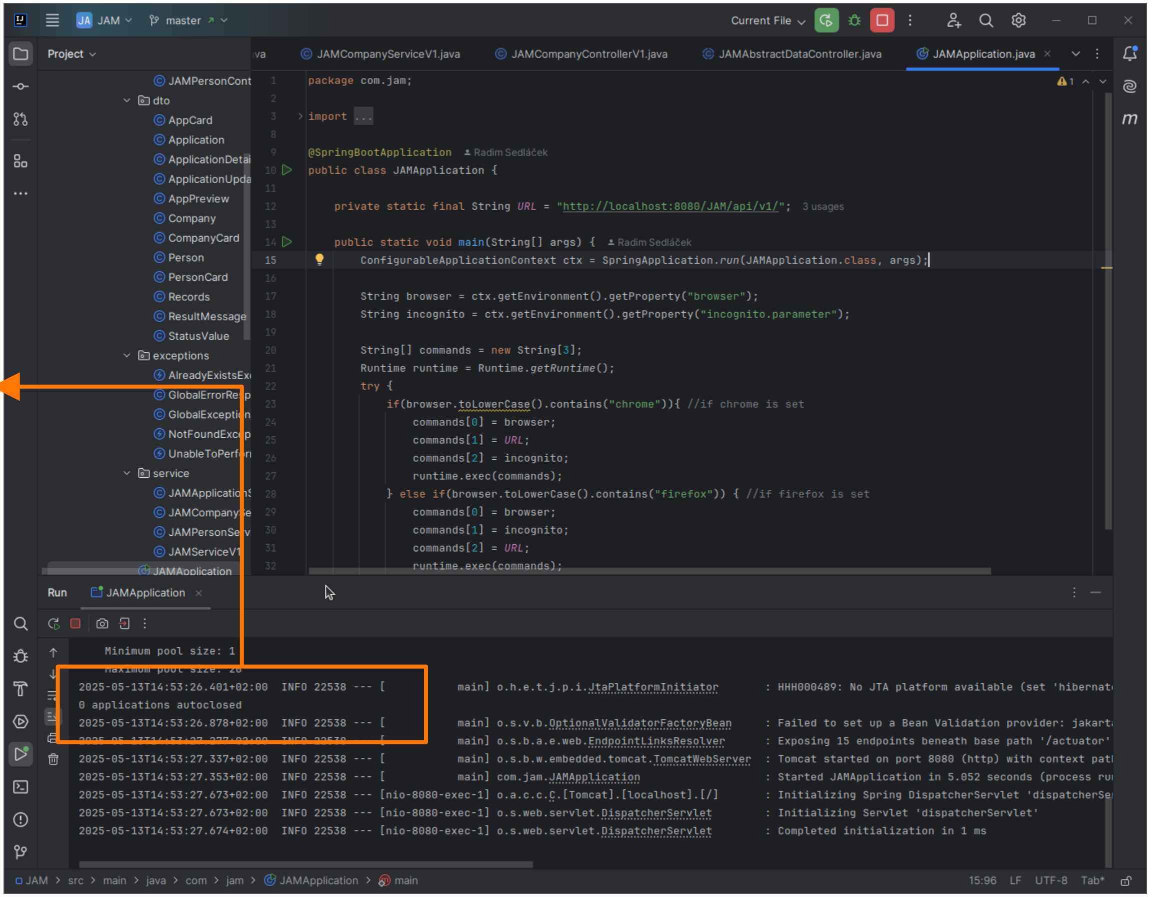
Hibernate static meta-model leveraged to enforce type safety

# Back-end layer

During every start the service checks database for application with idle time longer than 30 days and auto closes them



```
2025-04-16T14:18:55.220+02:00  INFO 18699 --- [
0 applications autoclosed
2025-04-16T14:18:55.458+02:00  INFO 18699 --- [
```

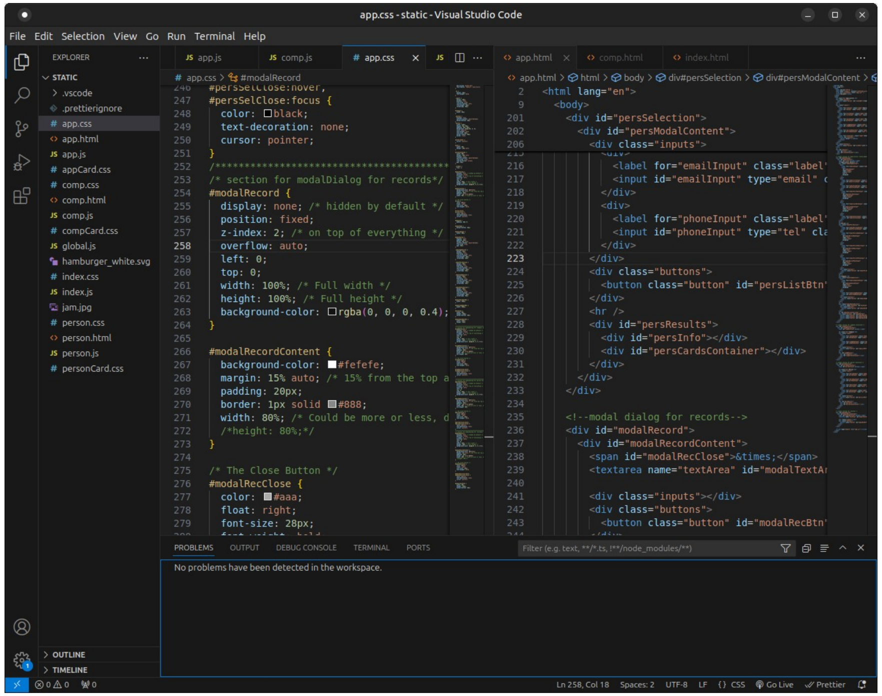


```java
package com.jam;

import ...

@SpringBootApplication  Radim Sedláček
public class JAMApplication {

    private static final String URL = "http://localhost:8080/JAM/api/v1/";  3 usages

    public static void main(String[] args) {  Radim Sedláček
        ConfigurableApplicationContext ctx = SpringApplication.run(JAMApplication.class, args);

        String browser = ctx.getEnvironment().getProperty("browser");
        String incognito = ctx.getEnvironment().getProperty("incognito.parameter");

        String[] commands = new String[3];
        Runtime runtime = Runtime.getRuntime();
        try {
            if(browser.toLowerCase().contains("chrome")){ //if chrome is set
                commands[0] = browser;
                commands[1] = URL;
                commands[2] = incognito;
                runtime.exec(commands);
            } else if(browser.toLowerCase().contains("firefox")) { //if firefox is set
                commands[0] = browser;
                commands[1] = incognito;
                commands[2] = URL;
                runtime.exec(commands);
```

```
    Minimum pool size: 1
2025-05-13T14:53:26.401+02:00  INFO 22538 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator       : HHH000489: No JTA platform available (set 'hibernat
0 applications autoclosed
2025-05-13T14:53:26.878+02:00  INFO 22538 --- [           main] o.s.v.b.OptionalValidatorFactoryBean     : Failed to set up a Bean Validation provider: jakart
                                                 [           main] o.s.b.a.e.web.EndpointLinksResolver      : Exposing 15 endpoints beneath base path '/actuator'
2025-05-13T14:53:27.337+02:00  INFO 22538 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8080 (http) with context pat
2025-05-13T14:53:27.353+02:00  INFO 22538 --- [           main] com.jam.JAMApplication                   : Started JAMApplication in 5.052 seconds (process ru
2025-05-13T14:53:27.673+02:00  INFO 22538 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherServlet 'dispatcherSe
2025-05-13T14:53:27.673+02:00  INFO 22538 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2025-05-13T14:53:27.674+02:00  INFO 22538 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 1 ms
```

# Front-end layer

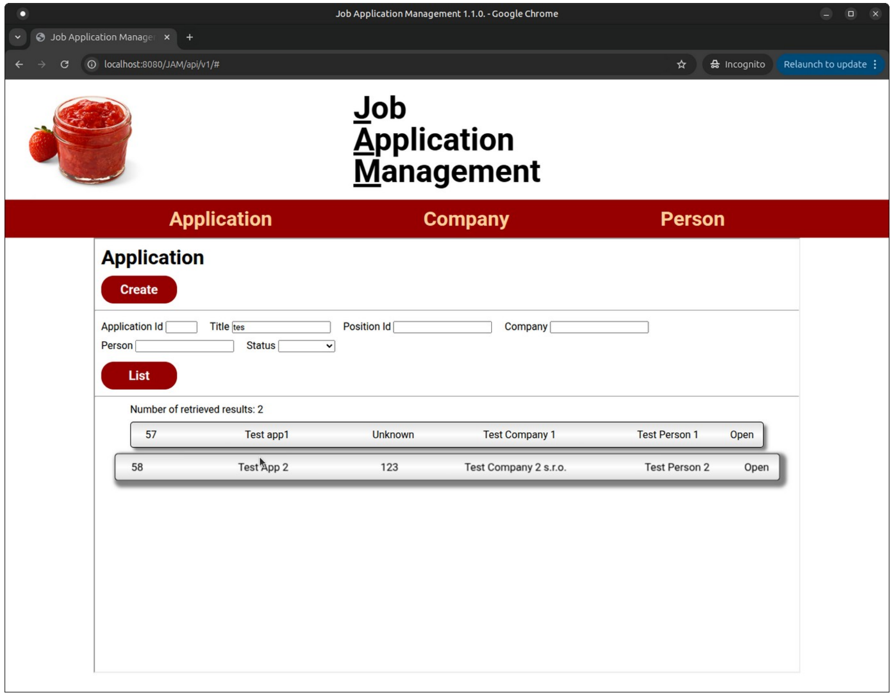
- HTML/CSS/JavaScript based web page
- Coded in Visual Studio Code



```css
#persSelClose:hover,
#persSelClose:focus {
  color: black;
  text-decoration: none;
  cursor: pointer;
}
/*****************************************
/* section for modalDialog for records*/
#modalRecord {
  display: none; /* hidden by default */
  position: fixed;
  z-index: 2; /* on top of everything */
  overflow: auto;
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  background-color: rgba(0, 0, 0, 0.4);
}

#modalRecordContent {
  background-color: #fefefe;
  margin: 15% auto; /* 15% from the top a
  padding: 20px;
  border: 1px solid #888;
  width: 80%; /* Could be more or less, d
  /*height: 80%;*/
}

/* The Close Button */
#modalRecClose {
  color: #aaa;
  float: right;
  font-size: 28px;
```

```html
<html lang="en">
  <body>
    <div id="persSelection">
      <div id="persModalContent">
        <div class="inputs">
            <label for="emailInput" class="label'
            <input id="emailInput" type="email" c
          </div>
          <div>
            <label for="phoneInput" class="label'
            <input id="phoneInput" type="tel" cla
          </div>
        </div>
        <div class="buttons">
          <button class="button" id="persListBtn'
        </div>
        <hr />
        <div id="persResults">
          <div id="persInfo"></div>
          <div id="persCardsContainer"></div>
        </div>
      </div>
    </div>

    <!--modal dialog for records-->
    <div id="modalRecord">
      <div id="modalRecordContent">
        <span id="modalRecClose">&times;</span>
        <textarea name="textArea" id="modalTextAr

        <div class="inputs"></div>
        <div class="buttons">
          <button class="button" id="modalRecBtn'
```

```
No problems have been detected in the workspace.
```


Job Application Management 1.1.0. - Google Chrome
Job Application Manage
localhost:8080/JAM/api/v1/#
Incognito
Relaunch to update
Job Application Management
Application
Company
Person
Create
List

Application

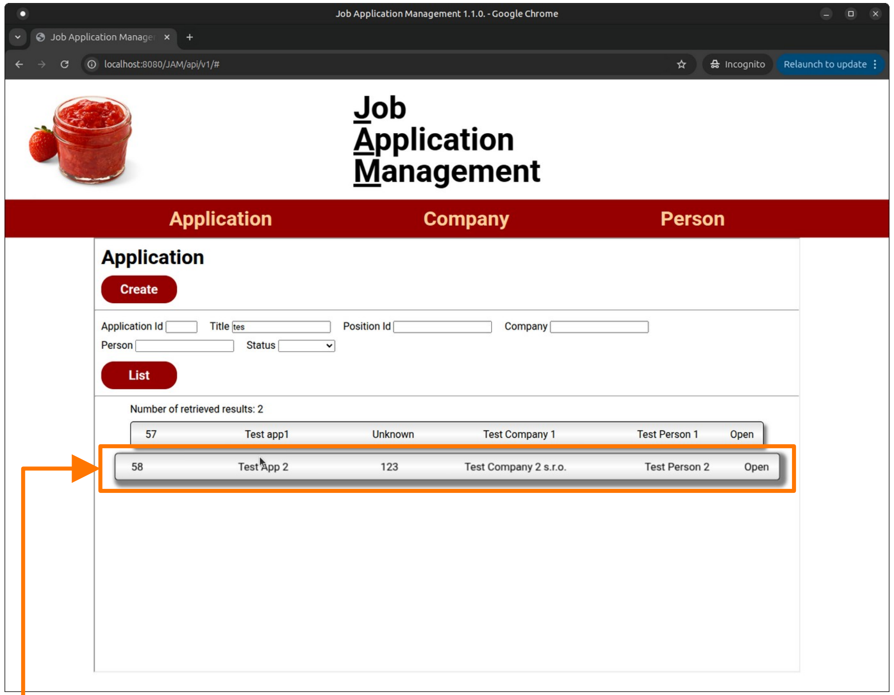
Application Id [ ] Title [tes] Position Id [ ] Company [ ]
Person [ ] Status [ ]

Number of retrieved results: 2

| | | | | | |
|---|---|---|---|---|---|
| 57 | Test app1 | Unknown | Test Company 1 | Test Person 1 | Open |
| 58 | Test App 2 | 123 | Test Company 2 s.r.o. | Test Person 2 | Open |

# Front-end layer

Responsive design for use on desktop, tablet as well as smart phone



Job Application Management 1.1.0. - Google Chrome

localhost:8080/JAM/api/v1/#

Job Application Management

Application | Company | Person

Application

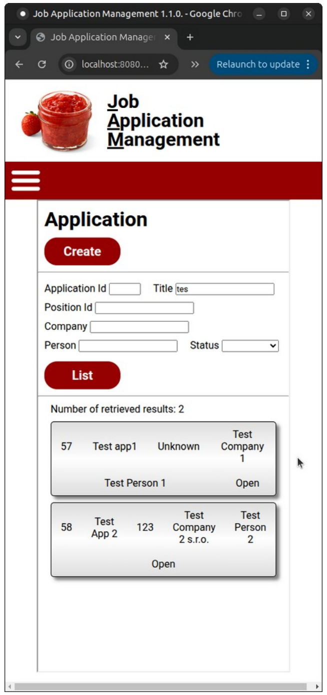
Create

Application Id [ ] Title [tes] Position Id [ ] Company [ ]
Person [ ] Status [ ]

List

Number of retrieved results: 2

| | | | | | |
|---|---|---|---|---|---|
| 57 | Test app1 | Unknown | Test Company 1 | Test Person 1 | Open |
| 58 | Test App 2 | 123 | Test Company 2 s.r.o. | Test Person 2 | Open |



Job Application Management 1.1.0. - Google Chro

Job Application Management

Application

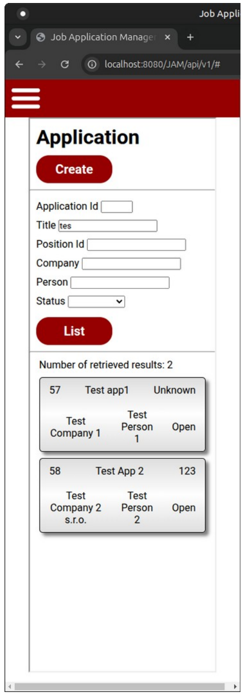
Create

Application Id [ ] Title [tes]
Position Id [ ]
Company [ ]
Person [ ] Status [ ]

List

Number of retrieved results: 2

| | | | | | |
|---|---|---|---|---|---|
| 57 | Test app1 | Unknown | Test Company 1 | Test Person 1 | Open |
| 58 | Test App 2 | 123 | Test Company 2 s.r.o. | Test Person 2 | Open |



localhost:8080/JAM/api/v1/#

Application

Create

Application Id [ ]
Title [tes]
Position Id [ ]
Company [ ]
Person [ ]
Status [ ]

List

Number of retrieved results: 2

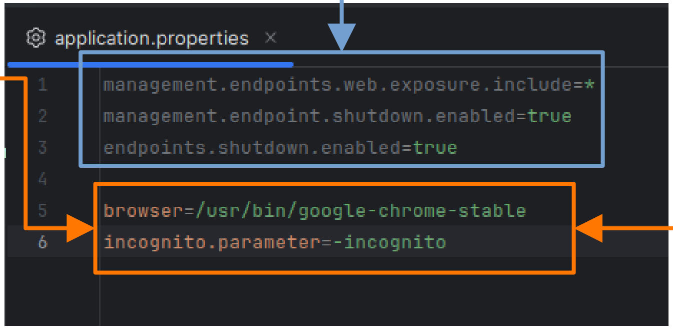
| | | | | | |
|---|---|---|---|---|---|
| 57 | Test app1 | Unknown | Test Company 1 | Test Person 1 | Open |
| 58 | Test App 2 | 123 | Test Company 2 s.r.o. | Test Person 2 | Open |

3D like effect when hovering over a card in the result list

# Front-end layer

Although created as a web app, currently is used as a desktop application.

Preferable web browser can be configured via application.properties (currently only Chrome and Firefox is supported). Is reccomended to start it in incognito/private mode. Browser starts automatically after application context is available.



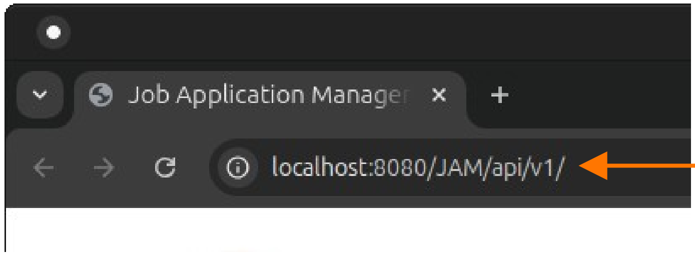
application.properties

```
management.endpoints.web.exposure.include=*
management.endpoint.shutdown.enabled=true
endpoints.shutdown.enabled=true

browser=/usr/bin/google-chrome-stable
incognito.parameter=-incognito
```

After leaving index.html page, the service is automatically stopped via spring actuator (currently works on Chrome only)

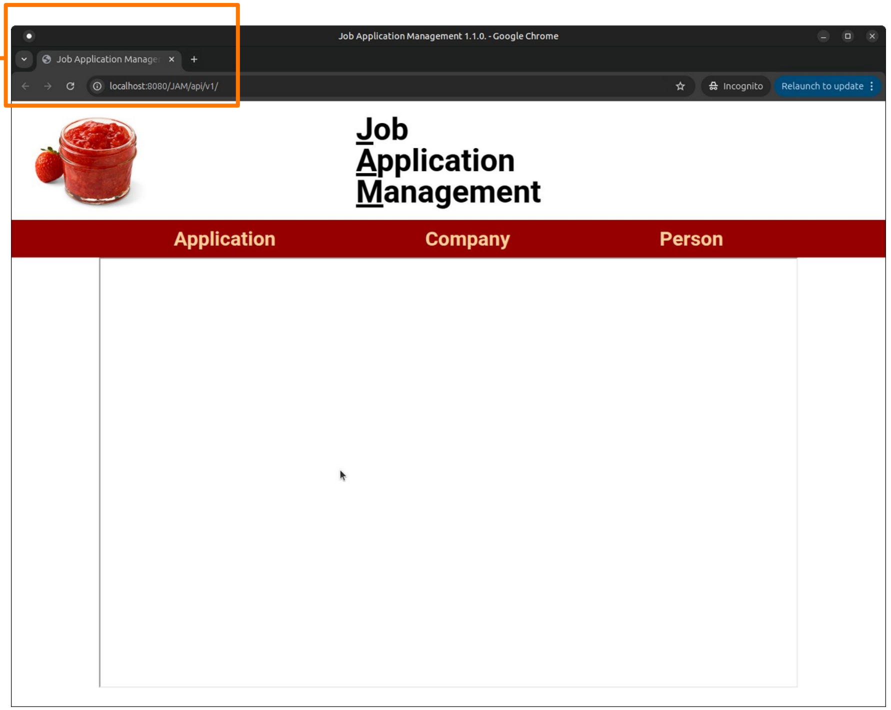
If unsupported or no browser is set, run it manually and enter address

# Front-end layer



Job Application Manager

localhost:8080/JAM/api/v1/



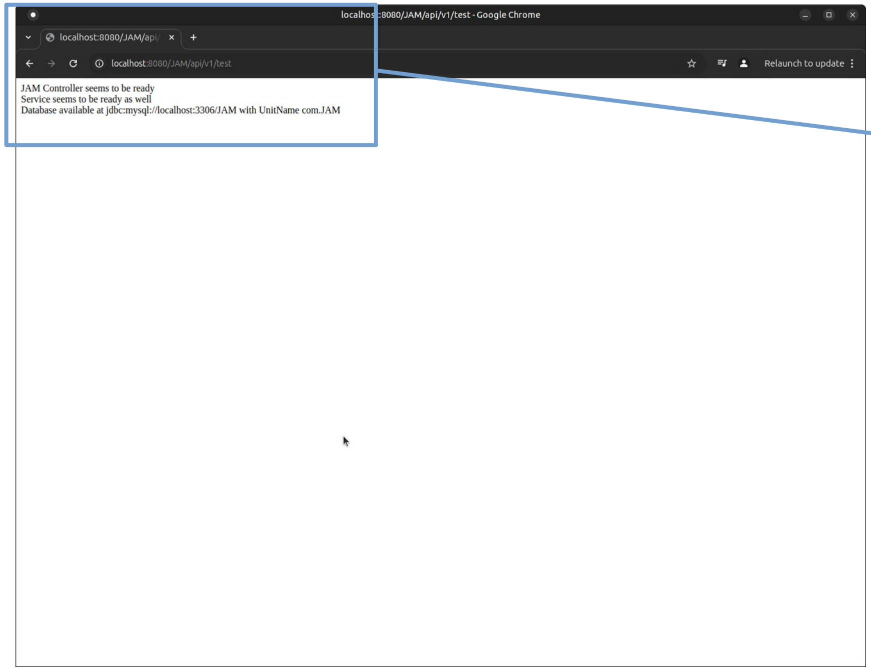
Job Application Management 1.1.0. - Google Chrome

Job Application Manage

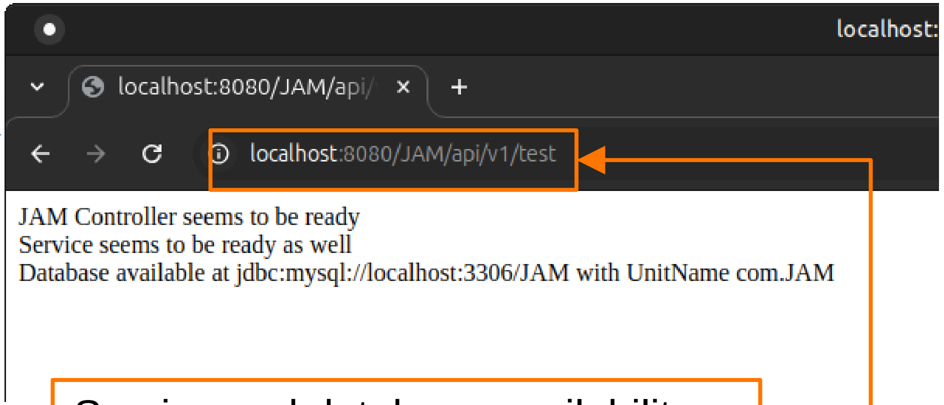
localhost:8080/JAM/api/v1/

When the server is running, the main web page itself available on “JAM/api/v1” address

# Front-end layer


localhost:8080/JAM/api/v1/test - Google Chrome
localhost:8080/JAM/api/
localhost:8080/JAM/api/v1/test
Relaunch to update

```
JAM Controller seems to be ready
Service seems to be ready as well
Database available at jdbc:mysql://localhost:3306/JAM with UnitName com.JAM
```



```
JAM Controller seems to be ready
Service seems to be ready as well
Database available at jdbc:mysql://localhost:3306/JAM with UnitName com.JAM
```

Service and database availability check accessible on “test” address

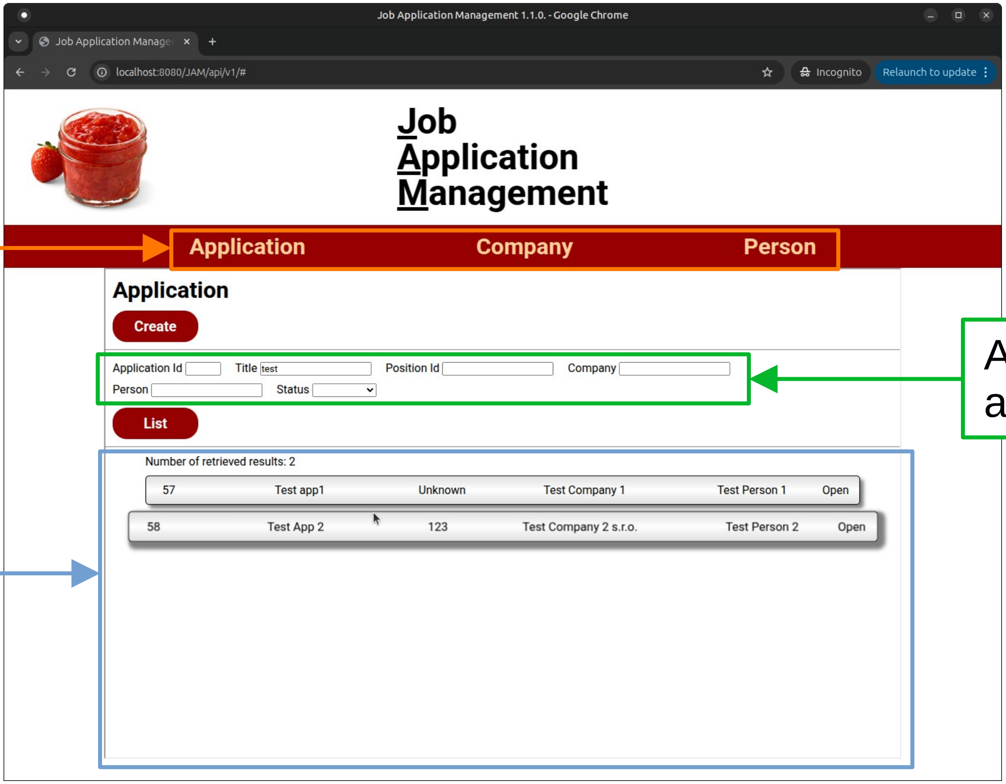
This test should be performed from new clean window, otherwise the service will stop due to unload of index.html page (see previous slides)

# Front-end layer

Area with entity to create/edit/delete selection

Area with filters available for each entity

Result list area



Job Application Management 1.1.0. - Google Chrome

localhost:8080/JAM/api/v1/#

Job Application Management

Application Company Person

Application

Create

Application Id  Title test  Position Id  Company

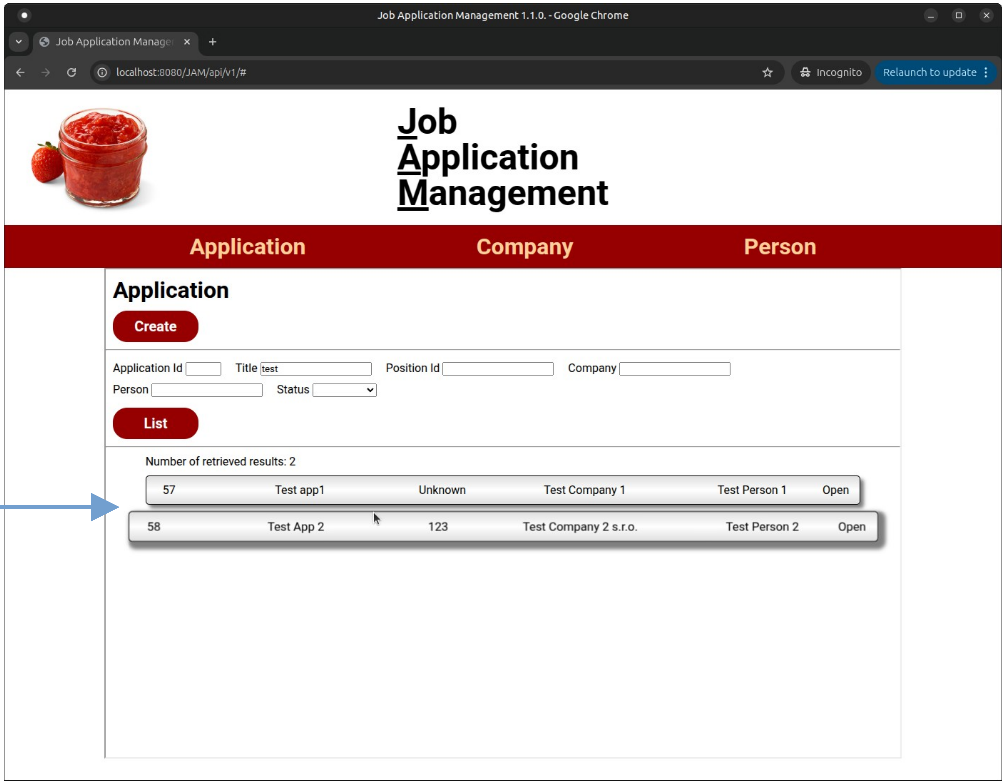
Person  Status

List

Number of retrieved results: 2

| | | | | | |
|---|---|---|---|---|---|
| 57 | Test app1 | Unknown | Test Company 1 | Test Person 1 | Open |
| 58 | Test App 2 | 123 | Test Company 2 s.r.o. | Test Person 2 | Open |

# Front-end layer



Job Application Management 1.1.0. - Google Chrome

Job Application Manager

localhost:8080/JAM/api/v1/#

Incognito

Relaunch to update

**Job Application Management**

Application    Company    Person

**Application**

Create

Application Id    Title test    Position Id    Company
Person    Status

List

Number of retrieved results: 2

| | | | | | |
|---|---|---|---|---|---|
| 57 | Test app1 | Unknown | Test Company 1 | Test Person 1 | Open |
| 58 | Test App 2 | 123 | Test Company 2 s.r.o. | Test Person 2 | Open |

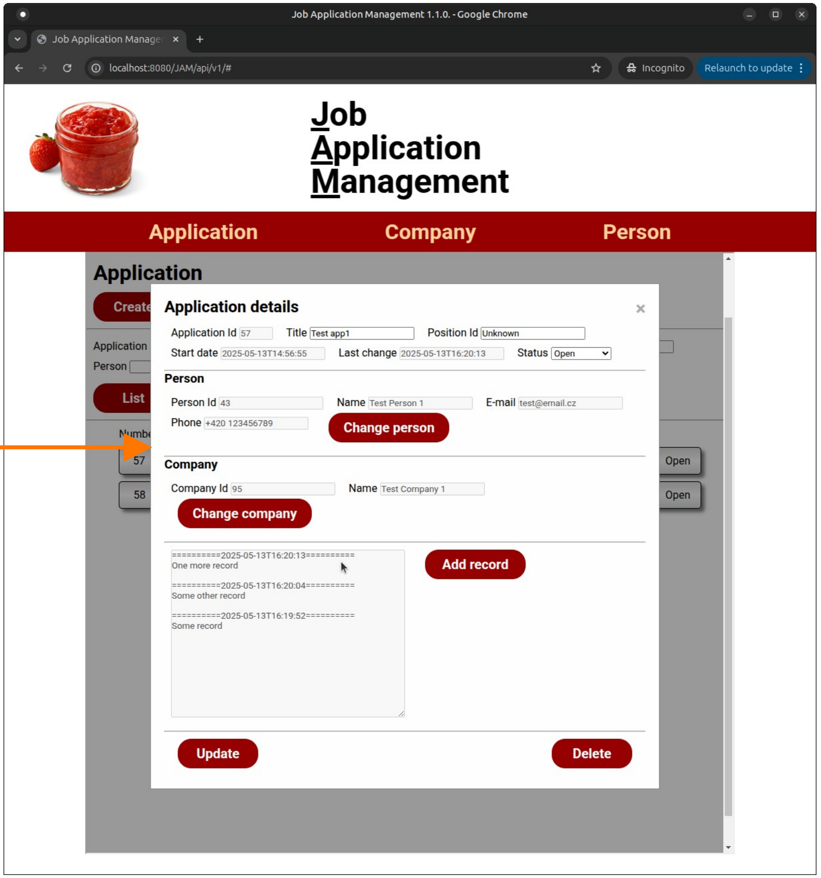
Data for each application card retrieved via “preview” view (see back-end layer section)

Clicking on a card will select given application for update

# Front-end layer

After selecting an application, modal window like dialog appears to allow changes.

Available changes include change of contact person or company as well as recording notes, or deletion of the application


Job Application Management 1.1.0. - Google Chrome

localhost:8080/JAM/api/v1/#

Application details

Application Id 57 Title Test app1 Position Id Unknown
Start date 2025-05-13T14:56:55 Last change 2025-05-13T16:20:13 Status Open

Person

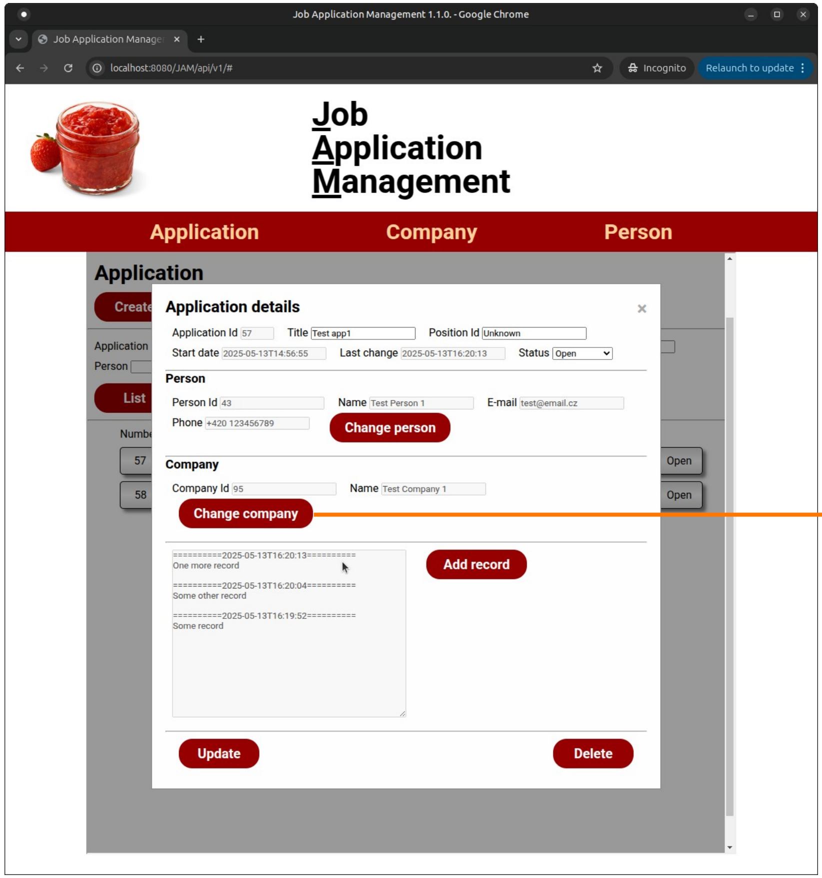
Person Id 43 Name Test Person 1 E-mail test@email.cz
Phone +420 123456789

Company

Company Id 95 Name Test Company 1

```
==========2025-05-13T16:20:13==========
One more record

==========2025-05-13T16:20:04==========
Some other record

==========2025-05-13T16:19:52==========
Some record
```

# Front-end layer



Job Application Management 1.1.0. - Google Chrome

localhost:8080/JAM/api/v1/#

**Job Application Management**

Application | Company | Person

## Application

**Application details**

Application Id 57  Title Test app1  Position Id Unknown
Start date 2025-05-13T14:56:55  Last change 2025-05-13T16:20:13  Status Open

**Person**

Person Id 43  Name Test Person 1  E-mail test@email.cz
Phone +420 123456789

Change person

**Company**

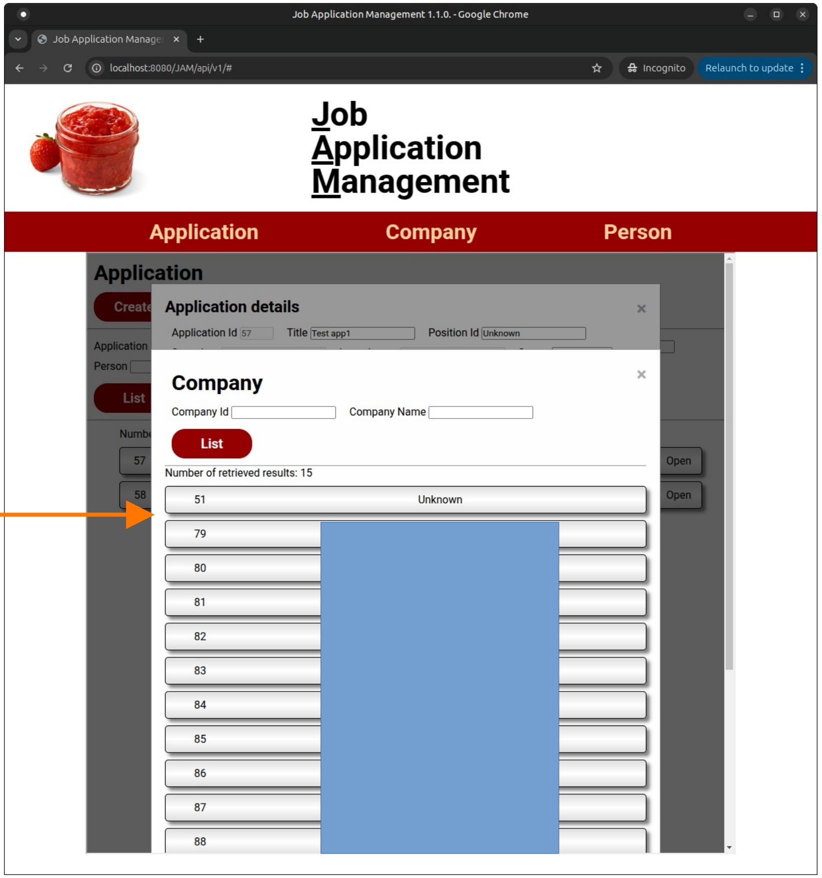
Company Id 95  Name Test Company 1

Change company

```
==========2025-05-13T16:20:13==========
One more record

==========2025-05-13T16:20:04==========
Some other record

==========2025-05-13T16:19:52==========
Some record
```

Add record

Update  Delete



Job Application Management 1.1.0. - Google Chrome

localhost:8080/JAM/api/v1/#

**Job Application Management**

Application | Company | Person

## Application

**Application details**

Application Id 57  Title Test app1  Position Id Unknown

**Company**

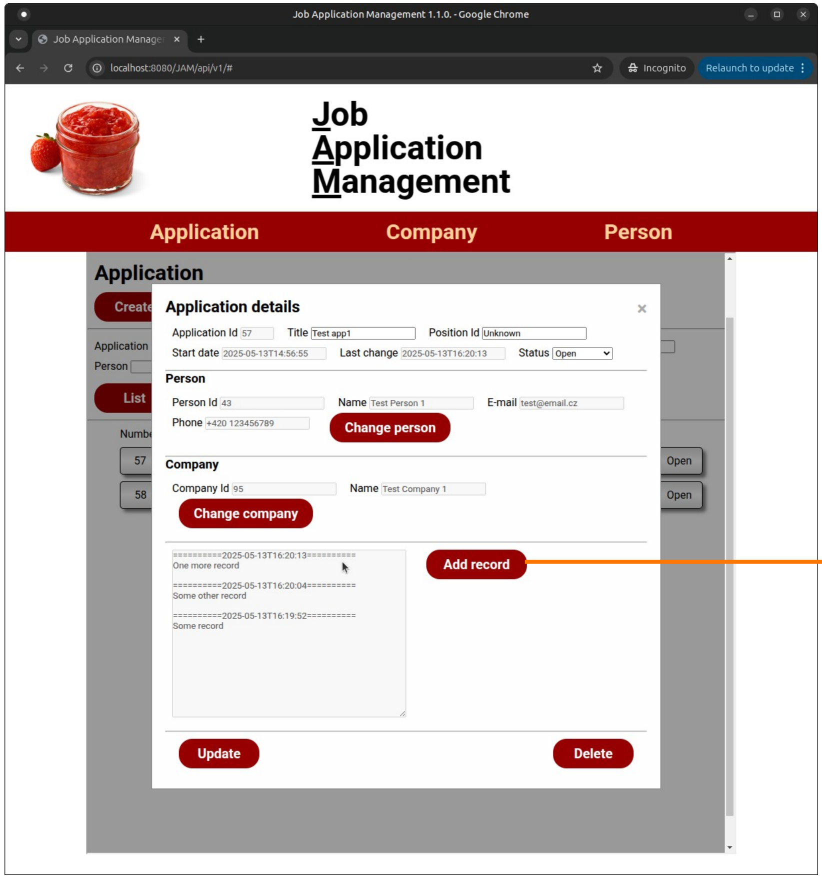
Company Id  Company Name

List

Number of retrieved results: 15

| | |
|---|---|
| 51 | Unknown |
| 79 | [redacted] |
| 80 | [redacted] |
| 81 | [redacted] |
| 82 | [redacted] |
| 83 | [redacted] |
| 84 | [redacted] |
| 85 | [redacted] |
| 86 | [redacted] |
| 87 | [redacted] |
| 88 | [redacted] |

# Front-end layer



Application

Application details

Application Id 57 Title Test app1 Position Id Unknown
Start date 2025-05-13T14:56:55 Last change 2025-05-13T16:20:13 Status Open

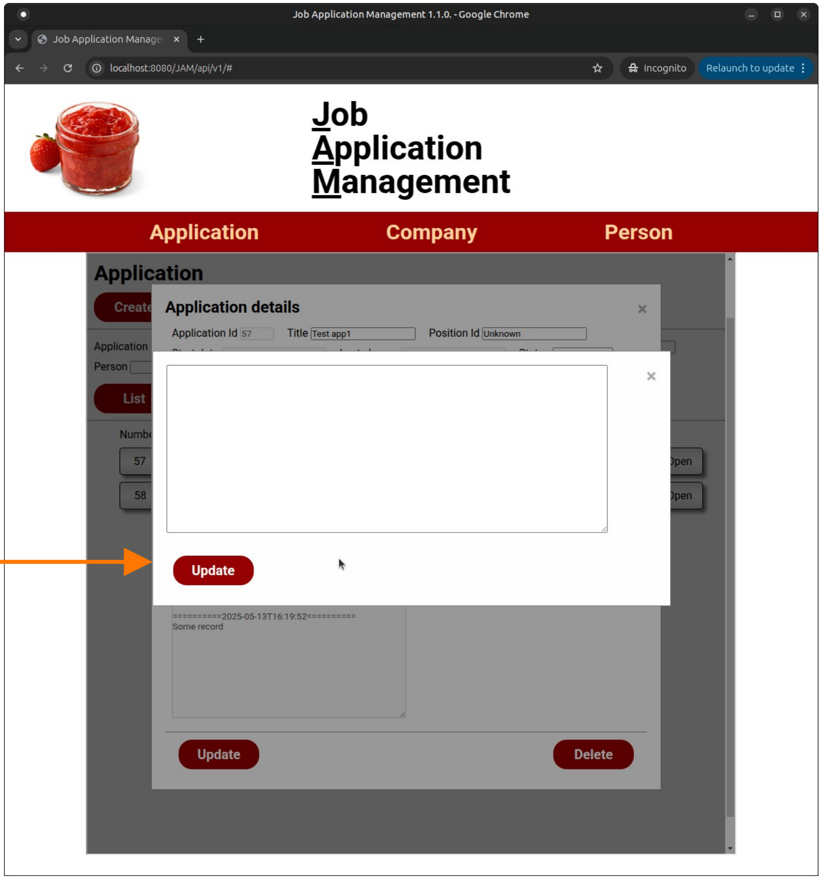
Person

Person Id 43 Name Test Person 1 E-mail test@email.cz
Phone +420 123456789

Company

Company Id 95 Name Test Company 1

```
=========2025-05-13T16:20:13=========
One more record

=========2025-05-13T16:20:04=========
Some other record

=========2025-05-13T16:19:52=========
Some record
```


Job Application Management 1.1.0. - Google Chrome
Job Application Manage
localhost:8080/JAM/api/v1/#
Incognito
Relaunch to update
Job Application Management
Application
Company
Person
Create
List
Update
Delete
Open

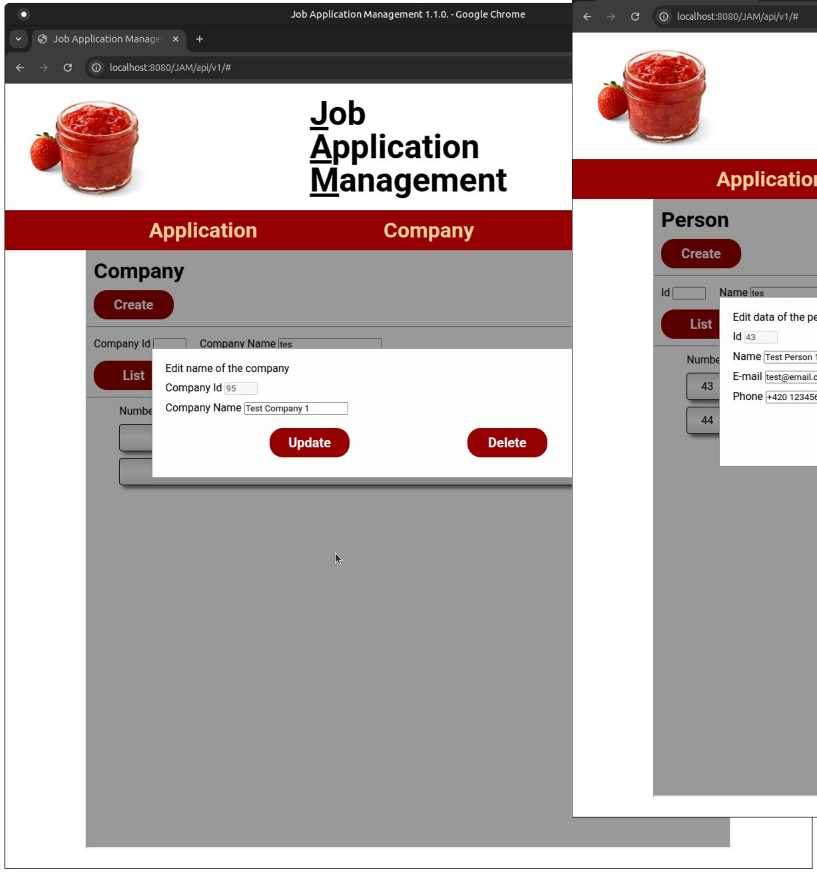
Application details

Application Id 57 Title Test app1 Position Id Unknown

```
=========2025-05-13T16:19:52=========
Some record
```

# Front-end layer

Same approach for editing an entity using modal like dialog is used for company and person as well


Job Application Management 1.1.0. - Google Chrome
localhost:8080/JAM/api/v1/#
Job Application Management
Application
Company
Create
List
Update
Delete

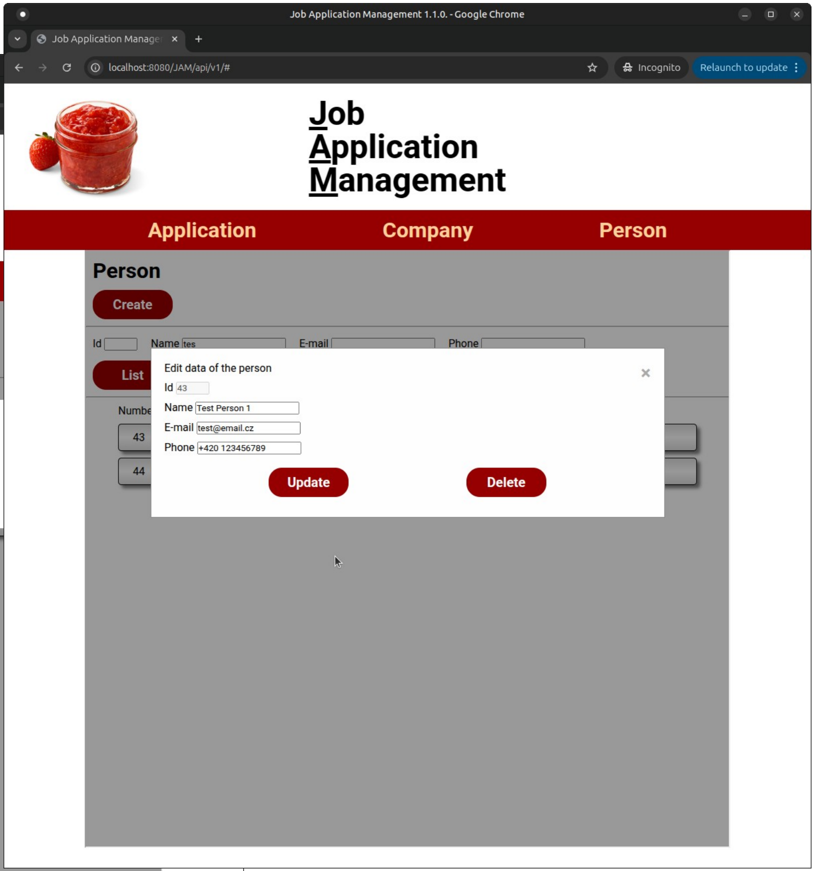
Company

Company Id Company Name tes

Edit name of the company

Company Id 95

Company Name Test Company 1


Job Application Management 1.1.0. - Google Chrome
localhost:8080/JAM/api/v1/#
Incognito
Relaunch to update
Job Application Management
Application
Company
Person
Create
List
Update
Delete

Person

Id Name tes E-mail Phone

Edit data of the person

Id 43

Name Test Person 1

E-mail test@email.cz

Phone +420 123456789

# JAM

Source code available on https://github.com/radimSed/JAM.git

Other work available on https://github.com/radimSed