Structured Data Management

Report

Stefan Bürscher - 1230486
Heinz Burgstaller - 1231266
14.06.2017

Motivation

Our application makes it possible to track information about politicians and their corruption history. The basic functions are: Add new politicians; Edit politicians and add a corruption suspicion. The index.html enables the user to see all possible actions.

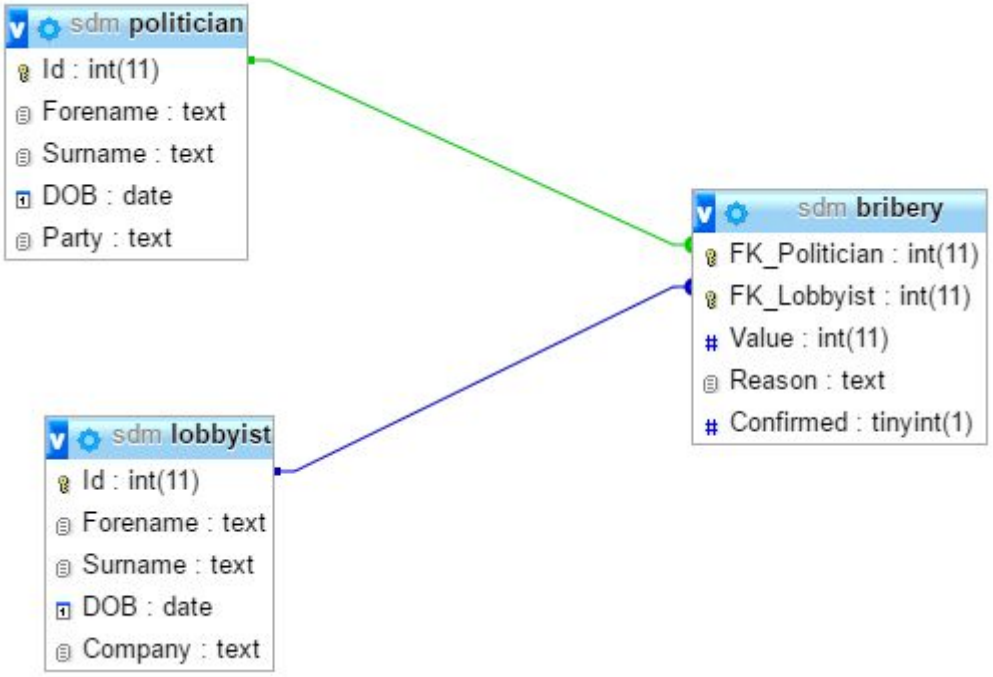
Installation

We developed with Java using Netbeans with web and Java EE plugins. We created a Java web servlet for handling GET and POST requests. The server we developed for is "Apache Tomcat 8.0.x". With the use of GSON[1] all server communication was done with JSON. The GUI is developed in HTML5, CSS3 and ECMAScript6. Google Chrome (>=59.x) is required to execute the GUI.

To deploy the server use the WAR file according to this guide[2]. The context should be accessible through port 8080.

To create the database import the SQL dump into your MySQL instance. Make sure the tables are accessible with user "sdm", password "sdm" and port 3306.

Database Schema in 3rd normal form



[1] https://github.com/google/gson
[2] https://tomcat.apache.org/tomcat-8.0-doc/deployer-howto.html

Practical implementation of the database with mySQL

```
-- phpMyAdmin SQL Dump
-- version 4.6.5.2
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Erstellungszeit: 13. Jun 2017 um 13:26
-- Server-Version: 10.1.21-MariaDB
-- PHP-Version: 5.6.30

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Datenbank: `sdm`
--
CREATE DATABASE IF NOT EXISTS `sdm` DEFAULT CHARACTER SET latin1 COLLATE latin1_swedish_ci;
USE `sdm`;

-- --------------------------------------------------------

--
-- Tabellenstruktur für Tabelle `bribery`
--

CREATE TABLE `bribery` (
  `FK_Politician` int(11) NOT NULL,
  `FK_Lobbyist` int(11) NOT NULL,
  `Value` int(11) NOT NULL,
  `Reason` text NOT NULL,
  `Confirmed` tinyint(1) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Daten für Tabelle `bribery`
--

INSERT INTO `bribery` (`FK_Politician`, `FK_Lobbyist`, `Value`, `Reason`, `Confirmed`)
VALUES
(0, 1, 10000, 'Murkraftwerk', 1),
(0, 2, 12000, 'Gesetz1', 1),
(0, 3, 15000, 'Gesetz2', 1),
(1, 2, 8000, 'Gesetz3', 1),
```

```
(2, 2, 4000, 'Gesetz4', 0),
(3, 1, 9000, 'Gesetz1', 1),
(3, 4, 6000, 'Gesetz13', 0),
(5, 4, 12000, 'Gesetz8', 1);

-- --------------------------------------------------------

--
-- Tabellenstruktur für Tabelle `lobbyist`
--

CREATE TABLE `lobbyist` (
  `Id` int(11) NOT NULL,
  `Forename` text NOT NULL,
  `Surname` text NOT NULL,
  `DOB` date NOT NULL,
  `Company` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Daten für Tabelle `lobbyist`
--

INSERT INTO `lobbyist` (`Id`, `Forename`, `Surname`, `DOB`, `Company`) VALUES
(1, 'Fritz', 'Phantom', '2017-06-11', 'Shell'),
(2, 'Tom', 'Turbo', '2016-05-11', 'BP'),
(3, 'Hans', 'Hermann', '2015-05-11', 'Estag'),
(4, 'Otto', 'Normalverbraucher', '2013-05-11', 'Monsanto'),
(5, 'Hans', 'Huber', '2012-02-10', 'Springer');

-- --------------------------------------------------------

--
-- Tabellenstruktur für Tabelle `politician`
--

CREATE TABLE `politician` (
  `Id` int(11) NOT NULL,
  `Forename` text NOT NULL,
  `Surname` text NOT NULL,
  `DOB` date NOT NULL,
  `Party` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Daten für Tabelle `politician`
--

INSERT INTO `politician` (`Id`, `Forename`, `Surname`, `DOB`, `Party`) VALUES
(0, 'Siegfried', 'Nagl', '1963-04-18', 'ÖVP'),
(1, 'Mario', 'Eustacchio', '1938-03-13', 'FPÖ'),
(2, 'Alexander', 'Van da Bellen', '1877-12-12', 'Die Grünen'),
(3, 'Wolfgang', 'Schüssel', '1950-02-12', 'ÖVP'),
(4, 'Jesus', 'Christus', '0000-01-01', 'KPÖ'),
```

```
(5, 'Alfred', 'Gusenbauer', '1957-12-04', 'SPÖ'),
(6, 'Michael', 'Ehmann', '1960-04-04', 'SPÖ'),
(7, 'Eva', 'Glawischnig', '1945-12-12', 'Die Grünen');

--
-- Indizes der exportierten Tabellen
--

--
-- Indizes für die Tabelle `bribery`
--
ALTER TABLE `bribery`
  ADD PRIMARY KEY (`FK_Politician`,`FK_Lobbyist`),
  ADD KEY `FK_Lobbyist` (`FK_Lobbyist`);

--
-- Indizes für die Tabelle `lobbyist`
--
ALTER TABLE `lobbyist`
  ADD PRIMARY KEY (`Id`);

--
-- Indizes für die Tabelle `politician`
--
ALTER TABLE `politician`
  ADD PRIMARY KEY (`Id`);

--
-- Constraints der exportierten Tabellen
--

--
-- Constraints der Tabelle `bribery`
--
ALTER TABLE `bribery`
  ADD CONSTRAINT `bribery_ibfk_1` FOREIGN KEY (`FK_Politician`) REFERENCES `politician`
(`Id`),
  ADD CONSTRAINT `bribery_ibfk_2` FOREIGN KEY (`FK_Lobbyist`) REFERENCES `lobbyist` (`Id`);

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

CREATE USER 'sdm'@'localhost' IDENTIFIED VIA mysql_native_password USING '***';
GRANT ALL PRIVILEGES ON *.* TO 'sdm'@'localhost' REQUIRE NONE WITH GRANT OPTION
MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0
MAX_USER_CONNECTIONS 0;
GRANT ALL PRIVILEGES ON `sdm`.* TO 'sdm'@'localhost';
```

Server-Side Scripts - Database Queries (PHP/servlets)

(All queries are provided in SDMServlet.java)

GET-Queries:

For searching we basically have three queries:

- get a list of all politicians
- get a list of all politicians + the sum of bribemoney
- get a politician by id

POST-Queries:

For updating and inserting data we have the following queries:

- insert new politician
- edit politician
- insert new suspicion of corruption

GET-Queries

Our GET-Queries are all called from our doGet-function:

```
@Override
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        response.setCharacterEncoding("UTF-8");

        String actionString = request.getParameter("action");
        GetAction action = null;
        try {
            action = GetAction.valueOf(actionString);
        } catch (IllegalArgumentException ex) {
            throw new RuntimeException("Specify GET action parameter! Possible: " +
Arrays.asList(GetAction.values()));
        }

        switch (action) {
            case getPoliticans:
                getPoliticans(response);
                break;
            case getPoliticanById:
                getPoliticanById(request, response);
                break;
            case getPoliticansWithMoney:
                getPoliticansWithMoney(response);
                break;
        }}
```

Get a chosen politician by id:

```
    private void getPoliticanById(HttpServletRequest request, HttpServletResponse response) throws
IOException {
        response.setContentType("application/json");
        Connection connection = null;
        PreparedStatement stmt = null;
        ResultSet rs = null;

        int politicanId = Integer.parseInt(request.getParameter("id"));

        try (PrintWriter pw = response.getWriter()) {
            connection = getDBConnection();
            stmt = connection.prepareStatement("select Id, Forename, Surname, Party, DOB from
politician where Id = ?");
            stmt.setInt(1, politicanId);
            stmt.execute();
```

```
            rs = stmt.getResultSet();
            Politican p = null;
            while (rs.next()) {
                int id = rs.getInt(1);
                String forename = rs.getString(2);
                String surname = rs.getString(3);
                String party = rs.getString(4);
                Date dob = rs.getDate(5);
                p = new Politican(id, forename, surname, party, dob, 0);
            }
            Gson gson = new GsonBuilder().setDateFormat("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'").create();
            String json = gson.toJson(p);
            pw.print(json);
        } catch (SQLException ex) {
            throw new ServerException("SQL", ex);
        } finally {
            try {
                if (rs != null) {
                    rs.close();
                }
            } catch (SQLException e) {
            }
            try {
                if (stmt != null) {
                    stmt.close();
                }
            } catch (SQLException e) {
            }
            try {
                if (connection != null) {
                    connection.close();
                }
            } catch (SQLException e) {
            }
        }
    }
```

get politicians and the amount of bribe money

```
private void getPoliticansWithMoney(HttpServletResponse response) throws IOException {
    response.setContentType("application/json");
    Connection connection = null;
    Statement stmt = null;
    ResultSet rs = null;

    try (PrintWriter pw = response.getWriter()) {
      connection = getDBConnection();
      stmt = connection.createStatement();
      String sql = "select p.Id, p.Forename, p.Surname, p.Party, p.DOB, coalesce(SUM(b.Value),0) "
              + " from politician p                                                              "
```

```
                + " left outer join bribery b on p.id = b.FK_Politician                                    "
                + " GROUP BY p.Id
    ";
        stmt.execute(sql);

        rs = stmt.getResultSet();
        List<Politican> politicans = new ArrayList<>();
        int value;
        while (rs.next()) {
          int id = rs.getInt(1);
          String forename = rs.getString(2);
          String surname = rs.getString(3);
          String party = rs.getString(4);
          Date dob = rs.getDate(5);
          value = rs.getInt(6);
          Politican p = new Politican(id, forename, surname, party, dob, value);
          politicans.add(p);
        }
        Gson gson = new GsonBuilder().setDateFormat("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'").create();
        String json = gson.toJson(politicans);

        pw.print(json);
      } catch (SQLException ex) {
        throw new ServerException("SQL", ex);
      } finally {
        try {
          if (rs != null) {
            rs.close();
          }
        } catch (SQLException e) {
        }
        try {
          if (stmt != null) {
            stmt.close();
          }
        } catch (SQLException e) {
        }
        try {
          if (connection != null) {
            connection.close();
          }
        } catch (SQLException e) {
        }
      }
    }
```

Get list of politicians:

```
private void getPoliticans(HttpServletResponse response) throws IOException {
    response.setContentType("application/json");
```

```
    Connection connection = null;
    Statement stmt = null;
    ResultSet rs = null;

    try (PrintWriter pw = response.getWriter()) {
        connection = getDBConnection();
        stmt = connection.createStatement();
        stmt.execute("select Id, Forename, Surname, Party, DOB from politician");

        rs = stmt.getResultSet();
        List<Politican> politicans = new ArrayList<>();
        while (rs.next()) {
            int id = rs.getInt(1);
            String forename = rs.getString(2);
                        String surname = rs.getString(3);
            String party = rs.getString(4);
            Date dob = rs.getDate(5);
            Politican p = new Politican(id, forename, surname, party, dob, 0);
            politicans.add(p);
        }
        Gson gson = new GsonBuilder().setDateFormat("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'").create();
        String json = gson.toJson(politicans);
        pw.print(json);
    } catch (SQLException ex) {
        throw new ServerException("SQL", ex);
    } finally {
        try {
            if (rs != null) {
                rs.close();
            }
        } catch (SQLException e) {
        }
        try {
            if (stmt != null) {
                stmt.close();
            }
        } catch (SQLException e) {
        }
        try {
            if (connection != null) {
                connection.close();
            }
        } catch (SQLException e) {
        }
    }
}
```

POST-Query

Our POST-Queries are all called from our doPost-function:

```
@Override
  public void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
    request.setCharacterEncoding("UTF-8");
    response.setCharacterEncoding("UTF-8");

    String actionString = request.getParameter("action");
    PostAction action = null;
    try {
      action = PostAction.valueOf(actionString);
    } catch (IllegalArgumentException ex) {
      throw new RuntimeException("Specify POST action parameter! Possible: " +
Arrays.asList(PostAction.values()));
    }

    switch (action) {
      case postPolitican:
        postPolitican(request, response);
        break;
      case postBribery:
        postBribery(request, response);
        break;
    }
  }
```

insert new politician/edit politician

```
private void postPolitican(HttpServletRequest request, HttpServletResponse response) throws
IOException {
    StringBuilder jb = new StringBuilder();
    BufferedReader reader = request.getReader();
    String line;
    while ((line = reader.readLine()) != null) {
      jb.append(line);
    }

    Gson gson = new Gson();
    Politican p = gson.fromJson(jb.toString(), Politican.class);

    Connection connection = null;
    PreparedStatement ps = null;
    String sql;
    if (p.getId() <= 0) {
      sql = "insert into politician"
              + " (Forename, Surname, Party, DOB) VALUES"
```

```
            + " (?, ?, ?, ?)";
    } else {
      sql = "update politician"
            + " set Forename = ?, Surname = ?, Party = ?, DOB = ?"
            + " where id = ?";
    }

    try {
      connection = getDBConnection();
      ps = connection.prepareStatement(sql);
      ps.setString(1, p.getForename());
      ps.setString(2, p.getSurname());
      ps.setString(3, p.getParty());
      ps.setDate(4, new java.sql.Date(p.getDob().getTime()));
      if (p.getId() > 0) {
        ps.setInt(5, p.getId());
      }
      ps.executeUpdate();
    } catch (SQLException ex) {
      throw new ServerException("SQL", ex);
    } finally {
      try {
        if (ps != null) {
          ps.close();
        }
      } catch (SQLException e) {
      }
      try {
        if (connection != null) {
          connection.close();
        }
      } catch (SQLException e) {
      }
    }

  }
```

insert new suspicion of corruption

```
private void postBribery(HttpServletRequest request, HttpServletResponse response) throws
IOException {
    StringBuilder jb = new StringBuilder();
    BufferedReader reader = request.getReader();
    String line;
    while ((line = reader.readLine()) != null) {
      jb.append(line);
    }

    Gson gson = new Gson();
    Bribery b = gson.fromJson(jb.toString(), Bribery.class);
```

```
  Connection connection = null;
  PreparedStatement ps = null;
  String sql = "insert into bribery"
          + " (FK_Politician, FK_Lobbyist, Value, Reason, Confirmed) VALUES"
          + " (?, ?, ?, ?, ?)";

  try {
    connection = getDBConnection();
    ps = connection.prepareStatement(sql);
    ps.setInt(1, b.getFkPolitician());
    ps.setInt(2, b.getFkLobbyist());
    ps.setInt(3, b.getValue());
    ps.setString(4, b.getReason());
    ps.setBoolean(5, b.isConfirmed());
    ps.executeUpdate();
  } catch (SQLException ex) {
    throw new ServerException("SQL", ex);
  } finally {
    try {
      if (ps != null) {
        ps.close();
      }
    } catch (SQLException e) {
    }
    try {
      if (connection != null) {
        connection.close();
      }
    } catch (SQLException e) {
    }
  }

}
```

Client-Side Scripts

- Index
    - List Politicians
    - Add Politician
    - Add corruption suspicion

Index

```
<html>
  <head>
    <title>SDM - Corruption Tracker</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
      input[type=text], input[type=date], input[type=number], select {
        width: 100%;
        padding: 12px 20px;
        margin: 8px 0;
        display: inline-block;
        border: 1px solid #ccc;
        border-radius: 4px;
        box-sizing: border-box;
      }

      input[type=button] {
        width: 100%;
        background-color: #4CAF50;
        color: white;
        padding: 14px 20px;
        margin: 8px 0;
        border: none;
        border-radius: 4px;
        cursor: pointer;
      }

      input[type=button]:hover {
        background-color: #45a049;
      }

      div {
        border-radius: 5px;
        background-color: #f2f2f2;
        padding: 20px;
      }
    </style>
  </head>
  <body>
```

```
    <h1>Corruption Tracker</h1>
    <div>
      <h3>Give us some hints for corrupt politicians and lobbyists...</h3>
      <input type="button" onclick="location.href = 'listPoliticiansHtml.html';"
value="List Politicians" />
      <input type="button" onclick="location.href = 'addPolitican.html';" value="Add
Politician" />
      <input type="button" onclick="location.href = 'addBribery.html';" value="Add Bribery"
/>
    </div>
  </body>
</html>
```

List Politicians

```
<html>
    <head>
        <style>
            table {
                border-collapse: collapse;
                width: 100%;
            }

            th, td {
                text-align: left;
                padding: 8px;
            }

            tr:nth-child(even){background-color: #f2f2f2}
        </style>
        <title>List Politicians</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">

    </head>
    <body>
        <div>
            <p> </p>
            <table style="width: 45px;" id="table_politicians">
                <tbody>
                    <tr id="titles">

                    </tr>
                </tbody>
            </table>
        </div>
        <script>

            var xhr = new XMLHttpRequest();
            xhr.open('GET', "http://localhost:8080/sdm/test?action=getPoliticansWithMoney",
true);
```

```
            xhr.send();

            xhr.addEventListener("readystatechange", processRequest, false);

            // Thats why!!!
            //xhr.onreadystatechange = processRequest;

            function processRequest(e) {

                //wird 2 mal ausgelöst !?!?
                if (xhr.readyState === 4 && xhr.status === 200) {
                    console.log(e);
                    var response = JSON.parse(xhr.responseText);

                    var row = document.getElementById("titles");
                    if (response.length > 0) {
                        for (var key in response[0]) {
                            var x = row.insertCell(-1);
                            x.innerHTML = key;
                        }
                        x = row.insertCell(-1);
                        x.innerHtml = " ";
                    }
                    var arrayLength = response.length;
                    for (var i = 0; i < arrayLength; i++) {
                        myFunction(response[i]);
                    }
                }
            }

            function editButtonPressed(x) {
                location.href="addPolitican.html?id="+x;
            }

            function myFunction(response) {
                var table = document.getElementById("table_politicians");
                var row = table.insertRow(-1);

                for (var key in response) {
                    console.log(key);
                    if (response.hasOwnProperty(key)) {

                        var val = response[key];
                        var cell1 = row.insertCell(-1);
                        cell1.innerHTML = val;
                    }
                }
                var cell = row.insertCell(-1);
                console.log(response);
                cell.innerHTML = "<button class=\"editbtn\"
onclick=\"editButtonPressed("+response.id+")\";>edit</button>";
            }

        </script>
```

```
    </body>
</html>
```

Add Politician

```
<html>
  <head>
    <title>Add a politican</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
      input[type=text], input[type=date], select {
        width: 100%;
        padding: 12px 20px;
        margin: 8px 0;
        display: inline-block;
        border: 1px solid #ccc;
        border-radius: 4px;
        box-sizing: border-box;
      }

      input[type=button] {
        width: 100%;
        background-color: #4CAF50;
        color: white;
        padding: 14px 20px;
        margin: 8px 0;
        border: none;
        border-radius: 4px;
        cursor: pointer;
      }

      input[type=button]:hover {
        background-color: #45a049;
      }

      div {
        border-radius: 5px;
        background-color: #f2f2f2;
        padding: 20px;
      }
    </style>
    <script>

      var url_string = window.location.href;
      var url = new URL(url_string);
      var c = url.searchParams.get("id");
      //console.log(c);
```

```
      if (c === null) {
        c = 0;
      }
      var xhr = new XMLHttpRequest();
      xhr.open('GET', "http://localhost:8080/sdm/test?action=getPoliticanById&id=" + c,
true);
      xhr.send();
      xhr.addEventListener("readystatechange", processRequest, false);


      function processRequest(e) {
        if (xhr.readyState === 4 && xhr.status === 200) {
          console.log(e);
          var response = JSON.parse(xhr.responseText);
          console.log(response);
          if (response !== null) {
            document.getElementById("fname").value = response.forename;
            document.getElementById("lname").value = response.surname;
            document.getElementById("party").value = response.party;
            document.getElementById("dob").value = response.dob.split('T')[0];
          }
        }
      }

      function savePolitican() {
        var xhr = new XMLHttpRequest();
        var url = "http://localhost:8080/sdm/test?action=postPolitican";
        xhr.open("POST", url, true);
        xhr.setRequestHeader("Content-type", "application/json");
        xhr.onreadystatechange = function () {
          if (xhr.readyState === 4 && xhr.status === 200) {
            //console.log(xhr.responseText);
            location.href = 'index.html';
          }
        };
        var dobValue = document.getElementById("dob").value;
        var dob = new Date(dobValue).toISOString();
        var partySelect = document.getElementById("party");
        var p = {
          "id": c,
          "forename": document.getElementById("fname").value,
          "surname": document.getElementById("lname").value,
          "party": partySelect.options[partySelect.selectedIndex].value,
          "dob": dob
        };
        xhr.send(JSON.stringify(p));
      }
    </script>
  </head>

  <body>

    <h3>Enter politican detail</h3>
```

```
    <div>
      <form>
        <label for="fname">First Name</label>
        <input type="text" id="fname" name="firstname">

        <label for="lname">Last Name</label>
        <input type="text" id="lname" name="lastname">

        <label for="party">Party</label>
        <select id="party" name="party">
          <option value="SPÖ">SPÖ</option>
          <option value="ÖVP">ÖVP</option>
          <option value="FPÖ">FPÖ</option>
          <option value="FPÖ">FPÖ</option>
          <option value="Die Grünen">Die Grünen</option>
          <option value="KPÖ">KPÖ</option>
          <option value="Piraten">Piraten</option>
        </select>

        <label for="dob">Date of birth</label>
        <input type="date" id="dob" name="dob">

        <input type="button" value="Submit" onclick="savePolitican()">
      </form>
    </div>

  </body>
</html>
```

Add corruption suspicion

```
<html>
  <head>
    <title>Add a bribery</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
      input[type=text], input[type=date], input[type=number], select {
        width: 100%;
        padding: 12px 20px;
        margin: 8px 0;
        display: inline-block;
        border: 1px solid #ccc;
        border-radius: 4px;
        box-sizing: border-box;
      }

      input[type=button] {
        width: 100%;
        background-color: #4CAF50;
```

```
      color: white;
      padding: 14px 20px;
      margin: 8px 0;
      border: none;
      border-radius: 4px;
      cursor: pointer;
    }

    input[type=button]:hover {
      background-color: #45a049;
    }

    div {
      border-radius: 5px;
      background-color: #f2f2f2;
      padding: 20px;
    }
  </style>
  <script>

    var xhr = new XMLHttpRequest();
    xhr.open('GET', "http://localhost:8080/sdm/test?action=getPoliticans", true);
    xhr.send();
    xhr.addEventListener("readystatechange", processRequest, false);

    function processRequest(e) {
      if (xhr.readyState === 4 && xhr.status === 200) {
        var response = JSON.parse(xhr.responseText);

        var pSelect = document.getElementById("politician");
        for (p in response) {
          //console.log(response[p]);
          var el = document.createElement("option");
          el.textContent = response[p].forename + " " + response[p].surname + ", " +
response[p].party;
          el.value = response[p].id;
          pSelect.appendChild(el);
        }

        xhr = new XMLHttpRequest();
        xhr.open('GET', "http://localhost:8080/sdm/test?action=getLobbyists", true);
        xhr.send();
        xhr.addEventListener("readystatechange", processRequest2, false);
      }
    }

    function processRequest2(e) {
      if (xhr.readyState === 4 && xhr.status === 200) {
        var response = JSON.parse(xhr.responseText);

        var lSelect = document.getElementById("lobbyist");
        for (l in response) {
          console.log(response[l]);
          let el = document.createElement("option");
```

```
            el.textContent = response[l].forename + " " + response[l].surname + ", " +
response[l].company;
            el.value = response[l].id;
            lSelect.appendChild(el);
          }
        }
      }

      function saveBribery() {
        var xhr = new XMLHttpRequest();
        var url = "http://localhost:8080/sdm/test?action=postBribery";
        xhr.open("POST", url, true);
        xhr.setRequestHeader("Content-type", "application/json");
        xhr.onreadystatechange = function () {
          if (xhr.readyState === 4 && xhr.status === 200) {
            //console.log(xhr.responseText);
            location.href = 'index.html';
          }
        };
        var pSelect = document.getElementById("politician");
        var lSelect = document.getElementById("lobbyist");
        var b = {
          "fkPolitician": pSelect.options[pSelect.selectedIndex].value,
          "fkLobbyist": lSelect.options[lSelect.selectedIndex].value,
          "value": document.getElementById("value").value,
          "reason": document.getElementById("reason").value,
          "confirmed": document.getElementById("con").checked
        };
        xhr.send(JSON.stringify(b));
      }
    </script>
  </head>

  <body>

    <h3>Enter bribery detail</h3>

    <div>
      <form>
        <label for="politician">Politician</label>
        <select id="politician"></select>

        <label for="lobbyist">Lobbyist</label>
        <select id="lobbyist"></select>

        <label for="value">Value</label>
        <input type="number" id="value">

        <label for="reason">Reason</label>
        <input type="text" id="reason">

        <label>
          <input type="checkbox" id="con">
          Confirmed
```

```
        </label>

        <input type="button" value="Submit" onclick="saveBribery()">
      </form>
    </div>

  </body>
</html>
```