

## Задание на лабораторную работу №4 Web-службы и REST

**Цель:** Разработать учебную подсистему ввода и модификации данных о клиентах условного сервиса используя метод взаимодействия компонентов распределённого приложения REST (Representational State Transfer — «передача репрезентативного состояния»).

**Теория.** См. материал лекции Web-службы и REST

**Среда.** OS - произвольно. Рекомендуемый язык программирования Java EE. Рекомендуемые базы данных - MySQL, Access (\*.mdb), MS SQL (одна любая, на выбор).

### Варианты задания.

1. Выбрать поля базы данных для ввода и хранения информации о клиентах для выполнения лабораторной работы согласно номеру варианта.

| Наименование поля      | Тип поля              | Обязательное (да, нет) | Вариант         |
|------------------------|-----------------------|------------------------|-----------------|
| Фамилия                | Текстовый             | Да                     | Любой           |
| Имя                    | Текстовый             | Да                     | Любой           |
| Отчество               | Текстовый             | Да                     | Любой           |
| Дата рождения          | Дата                  | Да                     | Любой           |
| Пол                    | Boolean               | Да                     | <b>Четный</b>   |
| Серия паспорта         | Текстовый             | Да                     | Любой           |
| № паспорта             | Текстовый (с маской)  | Да                     | Любой           |
| Кем выдан              | Текстовый             | Да                     | Любой           |
| Дата выдачи            | Дата                  | Да                     | Любой           |
| Идент. номер           | Текстовый (с маской)  | Да                     | <b>Четный</b>   |
| Место рождения         | Текстовый             | Да                     | Любой           |
| Город факт. проживания | Список (от 5 городов) | Да                     | Любой           |
| Адрес факт. проживания | Текстовый             | Да                     | Любой           |
| Телефон дом            | Текстовый (с маской)  | Нет                    | <b>Четный</b>   |
| Телефон моб            | Текстовый (с маской)  | Нет                    | Любой           |
| E-mail                 | Текстовый             | Нет                    | <b>Нечетный</b> |
| Место работы           | Текстовый             | Нет                    | <b>Нечетный</b> |
| Должность              | Текстовый             | Нет                    | <b>Нечетный</b> |

|                    |                       |     |                   |
|--------------------|-----------------------|-----|-------------------|
| Город прописки     | Список (от 5 городов) | Да  | 1,4,7... (N%3==1) |
| Адрес прописки     | Текстовый             | Да  | 2,5,8... (N%3==2) |
| Семейное положение | Список                | Да  | Любой             |
| Гражданство        | Список                | Да  | Любой             |
| Инвалидность       | Список                | Да  | Любой             |
| Пенсионер          | Boolean               | Да  | Любой             |
| Ежемесячный доход  | Денежный              | Нет | Любой             |
| Военнообязанный    | Boolean               | Да  | 3,6,9... (N%3==0) |

2. Разработать схему данных в которой для элементов типа *Список* предусмотрены отдельные справочники.

3. Реализовать схему данных в выбранной СУБД.

4. Реализовать API REST добавления, чтения, обновления и удаления (CRUD) клиентов в СУБД условного сервиса по адресу `http://host:port/rest/user/userID`, где `userID=0` для операции создания пользователя, и `id` для остальных операций. Ответ сервера содержит данные в формате:

- **json** (для вариантов 1..6, 13..18);
- **xml** (для вариантов 7..12, 19..24).

5. На хорошую оценку дополнительно реализовать API REST добавления, чтения, обновления и удаления (CRUD) группы клиентов в СУБД условного сервиса по адресу `http://host:port/rest/users`.

6. На отличную оценку предусмотреть серверную проверку корректности введенных значений в каждом из полей возвратом кода **400** – Bad request в случае ошибки или незаполнения обязательного поля.

7. Добавить в базу данных через разработанный программный интерфейс не менее 5 клиентов, включая себя.

8. На защите продемонстрировать четыре операции (CRUD).

### Рекомендации.

1. При проектировании внутренних Java-интерфейсов рекомендуется использовать паттерн Repository (см. <http://habrahabr.ru/post/263033/>).
2. Для проверки REST можно использовать расширение Postman или заранее заготовленные html-формы с вписанными по умолчанию данными.
3. Для возможности смены базы данных на более производительное решение подключение к базе данных удобнее выполнить через источник JDBC.
4. При подготовке работы на отличную оценку поля нужно проверить регулярным выражением, по смыслу соответствующим типу (или маске) поля, а поля справочников проверить на допустимость указанных в запросе значений.