



Кафедра вычислительной техники
Рефакторинг баз данных и приложений

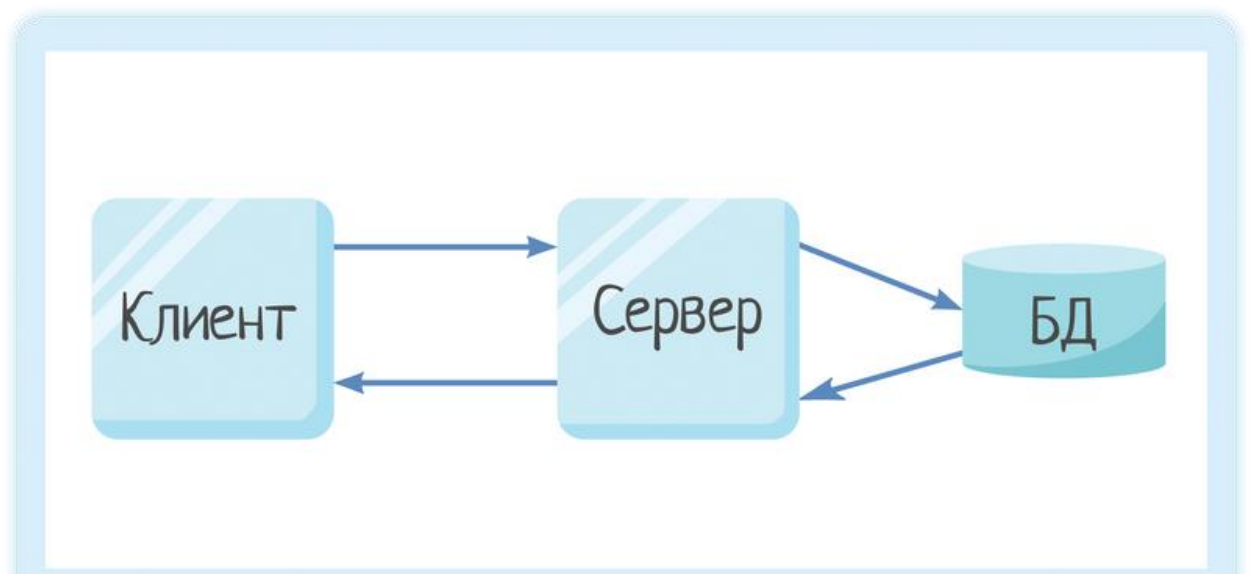
Рефакторинг мобильного приложения “Nomeragram”
Этап 2

Преподаватель:
Логинов Иван Павлович

Выполнили:
Полуянов Александр Михайлович
Девяткин Арсений Юрьевич
Р34141

Санкт-Петербург
2024

Была изменена архитектура приложения на клиент-серверную.



Были разработаны API

default ^

GET `/cars` Получить список автомобилей ^

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	Успешный ответ, возвращает список автомобилей	No links

Media type: Controls Accept header.

Example Value | Schema

```
[
  {
    "car_num": "string",
    "num_type": "individual",
    "year_of_issue": 0,
    "color": "string",
    "car_body_type": "sedan",
    "transmission": "variator",
    "wheel_drive_type": "front-wheel",
    "rudder": "left"
  }
]
```

default			^
GET	/cars	Получить список автомобилей	▼
POST	/cars	Добавить информацию об автомобиле	▼
GET	/cars/{car_num}	Получить информацию об автомобиле по номеру	▼
PUT	/cars/{car_num}	Обновить информацию об автомобиле	▼
DELETE	/cars/{car_num}	Удалить информацию об автомобиле	▼
GET	/photos	Получить список фотографий	▼
POST	/photos	Добавить фотографию	▼
GET	/insurance	Получить список страховок	▼
POST	/insurance	Добавить страховку	▼
GET	/crashes	Получить список аварий	▼
POST	/crashes	Добавить информацию об аварии	▼

С помощью OpenAPI была написана документация к API

```

openapi: 3.0.3
info:
  title: Nomeragram
  version: 1.0.0
  description: API для управления информацией о транспортных средствах, их
    владельцах, страховках, фотографиях и авариях.

servers:
  - url: https://nomeragram.com
    description: Production server

paths:
  /cars:
    get:
      summary: Получить список автомобилей
      responses:
        '200':
          description: Успешный ответ, возвращает список автомобилей
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/Car'
    post:
      summary: Добавить информацию об автомобиле
      requestBody:

```

```
    required: true
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Car'
  responses:
    '201':
      description: Автомобиль успешно добавлен
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Car'
    '400':
      description: Некорректные данные
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/ErrorResponse'

/cars/{car_num}:
  get:
    summary: Получить информацию об автомобиле по номеру
    parameters:
      - name: car_num
        in: path
        required: true
        schema:
          type: string
    responses:
      '200':
        description: Информация об автомобиле
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Car'
      '404':
        description: Автомобиль не найден
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ErrorResponse'
  put:
    summary: Обновить информацию об автомобиле
    parameters:
      - name: car_num
        in: path
        required: true
        schema:
          type: string
    requestBody:
      required: true
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Car'
    responses:
      '200':
        description: Информация об автомобиле обновлена
        content:
```

```
        application/json:
          schema:
            $ref: '#/components/schemas/Car'
      '400':
        description: Некорректные данные
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ErrorResponse'
delete:
  summary: Удалить информацию об автомобиле
  parameters:
    - name: car_num
      in: path
      required: true
      schema:
        type: string
  responses:
    '204':
      description: Автомобиль успешно удален
    '404':
      description: Автомобиль не найден
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/ErrorResponse'

/photos:
  get:
    summary: Получить список фотографий
    responses:
      '200':
        description: Успешный ответ, возвращает список фотографий
        content:
          application/json:
            schema:
              type: array
              items:
                $ref: '#/components/schemas/Photo'
  post:
    summary: Добавить фотографию
    requestBody:
      required: true
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Photo'
    responses:
      '201':
        description: Фотография успешно добавлена
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Photo'
      '400':
        description: Некорректные данные
        content:
          application/json:
            schema:
```

```

    $ref: '#/components/schemas/ErrorResponse'

/insurance:
  get:
    summary: Получить список страховок
    responses:
      '200':
        description: Успешный ответ, возвращает список страховок
        content:
          application/json:
            schema:
              type: array
              items:
                $ref: '#/components/schemas/Insurance'
  post:
    summary: Добавить страховку
    requestBody:
      required: true
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Insurance'
    responses:
      '201':
        description: Страховка успешно добавлена
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Insurance'
      '400':
        description: Некорректные данные
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ErrorResponse'

/crashes:
  get:
    summary: Получить список аварий
    responses:
      '200':
        description: Успешный ответ, возвращает список аварий
        content:
          application/json:
            schema:
              type: array
              items:
                $ref: '#/components/schemas/Crash'
  post:
    summary: Добавить информацию об аварии
    requestBody:
      required: true
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Crash'
    responses:
      '201':
        description: Информация об аварии успешно добавлена
```

```
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Crash'
'400':
  description: Некорректные данные
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/ErrorResponse'

components:
  schemas:
    Car:
      type: object
      properties:
        car_num:
          type: string
          description: Номер автомобиля
        num_type:
          type: string
          enum:
            - individual
            - legal entity
            - foreigner citizen
            - army
        year_of_issue:
          type: integer
          description: Год выпуска автомобиля
        color:
          type: string
          description: Цвет автомобиля
        car_body_type:
          type: string
          enum:
            - sedan
            - limousine
            - hatchback
            - liftback
            - universal
            - coupe
            - convertible
            - cabriolet
            - roadster
            - targa
            - minivan
            - pickup
            - truck
            - crossover
            - other
        transmission:
          type: string
          enum:
            - variator
            - mechanical
            - automatic
            - robotic
        wheel_drive_type:
          type: string
```

```
    enum:
      - front-wheel
      - rear-wheel
      - all-wheel
  rudder:
    type: string
    enum:
      - left
      - right
Photo:
  type: object
  properties:
    id:
      type: integer
      description: Уникальный идентификатор фотографии
    car_num:
      type: string
      description: Номер автомобиля, к которому относится фотография
    link:
      type: string
      description: Ссылка на фотографию
    date:
      type: string
      format: date
      description: Дата добавления фотографии
    added_by:
      type: integer
      description: Идентификатор пользователя, добавившего фотографию
Insurance:
  type: object
  properties:
    id:
      type: integer
      description: Уникальный идентификатор страховки
    car_num:
      type: string
      description: Номер автомобиля, на который оформлена страховка
    company:
      type: string
      description: Название страховой компании
    type:
      type: string
      description: Тип страховки
    amount:
      type: number
      format: float
      description: Страховая сумма
    price:
      type: number
      format: float
      description: Стоимость страховки
    start_date:
      type: string
      format: date
      description: Дата начала действия страховки
    end_date:
      type: string
      format: date
      description: Дата окончания действия страховки
```



```
Crash:
  type: object
  properties:
    id:
      type: integer
      description: Уникальный идентификатор аварии
    car_num:
      type: string
      description: Номер автомобиля, участвовавшего в аварии
    date:
      type: string
      format: date
      description: Дата аварии
    description:
      type: string
      description: Описание аварии
ErrorResponse:
  type: object
  properties:
    error:
      type: string
      description: Описание ошибки
```

Обновленный код: [ссылка](#)