Федеральное государственное автономное образовательное учреждение высшего образования

«**Университет ИТМО»**

**Факультет ПИиКТ**

**Дисциплина: Технологии веб-сервисов**

**Лабораторная работа 2**

Выполнили: Гурин Евгений Иванович, Камышанская Ксения Васильевна

Преподаватель: Дергачев Андрей Михайлович

Группа: Р4216

Санкт-Петербург 2024г.

# Задача

В данной работе в веб-сервис, разработанный в первой работе, необходимо добавить методы для создания, изменения и удаления записей из таблицы.

Метод создания должен принимать значения полей новой записи, метод изменения – идентификатор изменяемой записи, а также новые значения полей, а метод удаления – только идентификатор удаляемой записи.

Метод создания должен возвращать идентификатор новой записи, а методы обновления или удаления – статус операции. В данной работе следует вносить изменения только в standalone-реализацию сервиса.

В соответствии с изменениями сервиса необходимо обновить и клиентское приложение.

# Выполнение

Реализация происходила на основе кода, написанного в лабораторной 1. Сначала был создал модуль lab2, скопированный с lab1 и далее модифицировался.

1. Добавлены методы сохранения, модификации и удаления в интерфейс репозитория

|  |
| --- |
| /\*\*  \* Creates new item.  \*  \* @param entry Item to create  \* @return Created item  \*/  T create(@Valid T entry);  /\*\*  \* Update item.  \*  \* @param entry Item with updated fields.  \* @return Updated item  \* @throws EntryNotFound If item with specified id does not exist  \*/  T update(@Valid T entry) throws EntryNotFound;  /\*\*  \* Delete item by id.  \*  \* @param id Item id  \* @throws EntryNotFound If item with specified id does not exist  \*/  void deleteById(Integer id) throws EntryNotFound; |

1. Добавлены соответствующие методы в интерфейс АПИ сервиса

|  |
| --- |
| /\*\*  \* Create city.  \*  \* @param city new city data  \* @return created city  \*/  @WebMethod  CityDTO create(@WebParam(name = "city") CityCreateUpdateDTO city);  /\*\*  \* Update city.  \*  \* @param id id of city to update  \* @param city city data you want to update  \* @return updated city  \*/  @WebMethod  CityDTO update(@WebParam(name = "id") Integer id, @WebParam(name = "city") CityCreateUpdateDTO city)  throws EntryNotFound;  /\*\*  \* Update city part.  \*  \* @param id id of city to update  \* @param city city data you want to update  \* @return updated city  \*/  @WebMethod  CityDTO patch(@WebParam(name = "id") Integer id, @WebParam(name = "city") CityCreateUpdateDTO city)  throws EntryNotFound;  /\*\*  \* Delete city by id.  \*  \* @param id city id  \* @return delete status  \*/  @WebMethod  Boolean deleteById(@WebParam(name = "id") Integer id) throws EntryNotFound; |

1. Реализация методов репозитория

|  |
| --- |
| @Override  public T create(@Valid final T entry) {  try (EntityManager em = this.createEntityManager()) {  EntityTransaction tx = em.getTransaction();  tx.begin();  em.persist(entry);  tx.commit();  return entry;  }  }  @Override  public T update(@Valid final T entry) throws EntryNotFound {  try (EntityManager em = this.createEntityManager()) {  EntityTransaction tx = em.getTransaction();  tx.begin();  this.getById(entry.getId());  em.merge(entry);  tx.commit();  return entry;  }  }  @Override  public void deleteById(final Integer id) throws EntryNotFound {  try (EntityManager em = this.createEntityManager()) {  EntityTransaction tx = em.getTransaction();  tx.begin();  em.remove(this.getById(id));  tx.commit();  }  } |

1. Реализация методов сервиса

|  |
| --- |
| @Override  @WebMethod  @Action(  input = "http://service.guldilin/City/createRequest",  output = "http://service.guldilin/City/createResponse")  public CityDTO create(@WebParam(name = "city") final CityCreateUpdateDTO city) {  return this.cityRepository.create(city.mapToEntity()).mapToDTO();  }  @Override  @WebMethod  @Action(  input = "http://service.guldilin/City/updateRequest",  output = "http://service.guldilin/City/updateResponse",  fault = {  @FaultAction(  className = EntryNotFound.class,  value = "http://service.guldilin/City/update/Fault/EntryNotFound")  })  public CityDTO update(  @WebParam(name = "id") final Integer id, @WebParam(name = "city") final CityCreateUpdateDTO city)  throws EntryNotFound {  City cityEntry = this.cityRepository.getById(id);  city.updateEntity(cityEntry);  return this.cityRepository.update(cityEntry).mapToDTO();  }  @Override  @WebMethod  @Action(  input = "http://service.guldilin/City/patchRequest",  output = "http://service.guldilin/City/patchResponse",  fault = {  @FaultAction(  className = EntryNotFound.class,  value = "http://service.guldilin/City/patch/Fault/EntryNotFound")  })  public CityDTO patch(  @WebParam(name = "id") final Integer id, @WebParam(name = "city") final CityCreateUpdateDTO city)  throws EntryNotFound {  City cityEntry = this.cityRepository.getById(id);  city.patchEntity(cityEntry);  return this.cityRepository.update(cityEntry).mapToDTO();  }  @Override  @WebMethod  @Action(  input = "http://service.guldilin/City/deleteByIdRequest",  output = "http://service.guldilin/City/deleteByIdResponse")  public Boolean deleteById(@WebParam(name = "id") final Integer id) {  try {  this.cityRepository.deleteById(id);  return true;  } catch (Exception e) {  return false;  }  } |

1. Добавлены команды в консольный клиент

Пример для команды create

|  |
| --- |
| /\*\*  \* Executor for create method.  \*/  public class CreateExecutor extends Executor<CreateArgs> {  /\*\*  \* Create empty CreateArgs instance for parsing later.  \*  \* @return empty args object for executor  \*/  @Override  public CreateArgs createEmptyArgs() {  return new CreateArgs();  }  /\*\*  \* Execute create command.  \*  \* @param argv Unparsed CLI args  \* @param args Parsed CLI args  \* @param serviceProvider ServiceProvider implementation  \*/  @Override  public void execute(final String[] argv, final CreateArgs args, final ServiceProvider serviceProvider) {  var cityService = serviceProvider.provideCityService();  var created = cityService.create(args.toDTO());  EntitiesPrinter.print(System.out, created);  }  } |

Итоговый интерфейс консольных команд

Usage: lab3-client.jar [options]

Options:

\* -area

City area number

\* -car-code

City car code number

\* -c, -command

Command name

Possible Values: [find, create, update, patch, delete]

-help, -h

Show help message (also can be used with -c argument to show help

message for command)

\* -meters-above-sea-level

Meters above sea level for City

\* -name

City name

\* -population

City population value

\* -population-density

City population density

\* -url

Server URL base (example: http://localhost:8080)

# Вывод

В результате лабораторной работы были добавлены методы создания, изменения и удаления элементов в репозиторий и сервис. Кроме того был обновлен консольный клиент.