

# Проект “Уточка”

## 1. Описание

Уточка, изготовленная из любого материала любого цвета любой высоты.

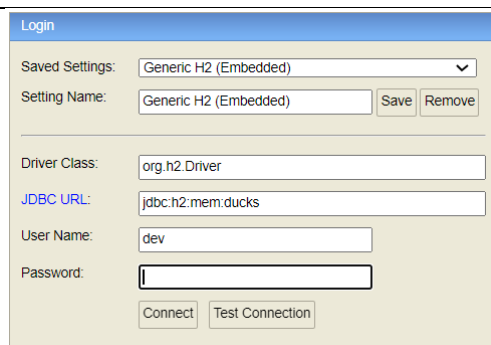
Может плавать, летает (если ей не связали крылья) и крикает по команде заданное количество раз с заданной периодичностью.

## 2. Требования к реализации

### 2.1. Данные о сервисе

Сервис:	duck-service
URL:	http://localhost:2222
	http://localhost:2222/swagger-ui.html
JAR:	duck-service-1.0.0.jar
Строка запуска сервиса:	<code>java -jar duck-service-1.0.0.jar</code>
Консоль для работы с базой данных	http://localhost:2222/h2-console

Настройки соединения с базой данных



Пароль - dev

### 2.2. Характеристики объекта

Объект: Уточка

Характеристики:

Характеристика	Поле	Значение
Цвет	color	любое
Материал	material	любой
Размер	height	любой
Звук	sound	quack
Состояние крыльев	wingsState	ACTIVE, FIXED

## 2.3. Методы

- Создать
- Обновить
- Удалить
- Получить список идентификаторов
- Получить характеристики уточки
- Крякать
- Лететь
- Плыть

### 2.3.1. Метод create

Метод позволяет создать объект и сохранить его в БД.

Тип	API	Параметры	Результат
POST	/api/duck/create	<pre>{   "color": "string",   "height": 0,   "material": "string",   "sound": "string",   "wingsState": "ACTIVE" }</pre>	HTTP status code: 200 OK Body: <pre>{     "color": "string",     "height": 0,     "material": "string",     "sound": "string",     "wingsState": "ACTIVE"   }</pre>

### 2.3.2. Метод update

Тип	API	Параметры	Результат																										
PUT	/api/duck/update	<div>Parameters</div> <table><tr><th>Name</th><th>Description</th></tr><tr><td><b>color</b> * required string (query)</td><td>Цвет уточки</td></tr><tr><td></td><td><input type="text" value="yellow"/></td></tr><tr><td><b>height</b> * required number(\$double) (query)</td><td>Высота уточки</td></tr><tr><td></td><td><input type="text" value="0.01"/></td></tr><tr><td><b>id</b> * required integer(\$int64) (query)</td><td>Идентификатор уточки</td></tr><tr><td></td><td><input type="text" value="1"/></td></tr><tr><td><b>material</b> * required string (query)</td><td>Материал уточки</td></tr><tr><td></td><td><input type="text" value="rubber"/></td></tr><tr><td><b>sound</b> * required string (query)</td><td>Звук уточки</td></tr><tr><td></td><td><input type="text" value="quack"/></td></tr><tr><td>wingsState string (query)</td><td>wingsState</td></tr><tr><td></td><td><div>ACTIVE</div></td></tr></table>	Name	Description	<b>color</b> * required string (query)	Цвет уточки		<input type="text" value="yellow"/>	<b>height</b> * required number(\$double) (query)	Высота уточки		<input type="text" value="0.01"/>	<b>id</b> * required integer(\$int64) (query)	Идентификатор уточки		<input type="text" value="1"/>	<b>material</b> * required string (query)	Материал уточки		<input type="text" value="rubber"/>	<b>sound</b> * required string (query)	Звук уточки		<input type="text" value="quack"/>	wingsState string (query)	wingsState		<div>ACTIVE</div>	HTTP status code: 200 OK Body: { "message": "Duck with id = 1 is updated" }
Name	Description																												
<b>color</b> * required string (query)	Цвет уточки																												
	<input type="text" value="yellow"/>																												
<b>height</b> * required number(\$double) (query)	Высота уточки																												
	<input type="text" value="0.01"/>																												
<b>id</b> * required integer(\$int64) (query)	Идентификатор уточки																												
	<input type="text" value="1"/>																												
<b>material</b> * required string (query)	Материал уточки																												
	<input type="text" value="rubber"/>																												
<b>sound</b> * required string (query)	Звук уточки																												
	<input type="text" value="quack"/>																												
wingsState string (query)	wingsState																												
	<div>ACTIVE</div>																												

### 2.3.3.Метод delete

Тип	API	Параметры	Результат
DELETE	/api/duck/delete	Идентификатор уточки	HTTP status code: 200 OK Body: { "message": "Duck is deleted" }

### 2.3.4.Метод getAllIds

Тип	API	Параметры	Результат
GET	/api/duck/getAllIds	нет	HTTP status code: 200 OK Body(список существующих в БД id, например): [1,2,3]

### 2.3.5.Метод properties

Метод позволяет получить характеристики Уточка.

Тип	API	Параметры	Результат
POST	/api/duck/action/properties	Идентификатор уточки	HTTP status code: 200 OK Body: { "color": "string", "height": 0, "material": "string", "sound": "string", "wingsState": "ACTIVE" }

### 2.3.6.Метод fly

Метод позволяет Уточке лететь или отказаться от полёта.

Тип	API	Параметры	Результат
GET	/api/duck/action/fly	Идентификатор Уточка	Если wingsState=ACTIVE, то HTTP status code: 200 OK Body: { "message": "I'm flying" }  Если wingsState=FIXED , то HTTP status code: 200 OK Body: { "message": "I can't fly" }

### 2.3.7.Метод swim

Метод позволяет Уточке плыть.

Тип	API	Параметры	Результат
GET	/api/duck/swim	Идентификатор Уточка	HTTP status code: 200 OK Body: { "message": "I'm swimming" }

## 2.3.8.Метод quack

Метод позволяет Уточке крякать.

Тип	API	Параметры	Результат
POST	/api/duck/action/quack	Идентификатор Уточки. Количество повторений, количество «кряков» в звуке передаются в URL. Описание см. по тексту ниже (Параметры метода)	Описание см. по тексту ниже (Возвращаемый результат)

### Параметры метода

Кроме идентификатора Уточка, параметрами являются количество повторений, количество «кряков» в звуке.

Особенности:

- количество повторений не может быть < 0, может быть = 0, целое число
- количество «кряков» в звуке не может быть < 0, может быть = 0, целое число

### Возвращаемый результат

HTTP status code: 200 OK

body (например, количество повторений=3, количество «кряков» в звуке=2):

```
{ "sound": "quack-quack, quack-quack, quack-quack" }
```

Особенности:

- если количество «кряков» == 0, то HTTP status code - 200 OK и body - { "sound": "" }
- если количество повторений == 0, то 200 OK и body - { "sound": " " }