

Homework_Lesson 13

Можно на выбор взять сервис:

OpenWeather (<https://openweathermap.org/>)

StarWars (<https://swapi.dev/>)

1. Через функцию составить запрос к API (внимательно посмотрите документацию сервиса), получить данные.
2. Составить функцию, которая будет забирать данные вашего запроса и представлять их в читаемом виде. После того как вы запишите это в виде скрипта, запустите его, вы сможете вызывать вашу функцию наподобие командлета.

Установим curl и jq для обработки JSON:

```
17:14:41 x 221 ~ sudo apt install curl jq
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done

17:14:52 222 ~ mkdir hw13
mkdir: created directory 'hw13'
```

Теперь создадим Bash-скрипт, который будет: отправлять запрос к API. Получать данные. Обрабатывать и выводить их в читаемом виде.

```
#!/bin/bash

# URL API для получения информации о персонажах (например, Люк Скайуокер)
API_URL="https://swapi.dev/api/people/1/" # Люк Скайуокер

# Отправляем GET-запрос к API и получаем ответ в формате JSON
RESPONSE=$(curl -s "$API_URL")

# Проверяем, если запрос успешен
if [[ $? -ne 0 ]]; then
    echo "Ошибка при запросе данных с API."
    exit 1
fi

# Используем jq для обработки JSON и извлекаем читаемую информацию
NAME=$(echo "$RESPONSE" | jq -r '.name')
HEIGHT=$(echo "$RESPONSE" | jq -r '.height')
MASS=$(echo "$RESPONSE" | jq -r '.mass')
HAIR_COLOR=$(echo "$RESPONSE" | jq -r '.hair_color')
SKIN_COLOR=$(echo "$RESPONSE" | jq -r '.skin_color')
BIRTH_YEAR=$(echo "$RESPONSE" | jq -r '.birth_year')
GENDER=$(echo "$RESPONSE" | jq -r '.gender')

# Выводим информацию в читаемом виде
echo "Информация о персонаже: $NAME"
echo "Рост: $HEIGHT см"
echo "Масса: $MASS кг"
echo "Цвет волос: $HAIR_COLOR"
echo "Цвет кожи: $SKIN_COLOR"
echo "Год рождения: $BIRTH_YEAR"
echo "Пол: $GENDER"
```

Сделаем файл скрипта исполняемым и запустим

```
17:28:24 224 ~ chmod +x sw.sh
17:28:26 225 ~ ./sw.sh
Информация о персонаже: Luke Skywalker
Рост: 172 см
Масса: 77 кг
Цвет волос: blond
Цвет кожи: fair
Год рождения: 19BBY
Пол: male
```

3. Добавить один (*два) входных параметра (абсолютно любой, например, название города для сервиса погоды либо номера страницы из массива персонажей)

Для этого создадим второй Bash-скрипт

```
#!/bin/bash

# Проверка на наличие параметров
if [ $# -lt 2 ]; then
    echo "Usage: $0 <category> <id>"
    echo "Categories: people, planets, films"
    exit 1
fi

# Входные параметры
CATEGORY="$1" # Категория: people, planets, films
ID="$2"        # ID в указанной категории

# Формирование URL
API_URL="https://swapi.dev/api/$CATEGORY/$ID/"

# Отправляем запрос к API и получаем ответ в формате JSON
RESPONSE=$(curl -s "$API_URL")

# Проверяем, если запрос успешен
if [ $? -ne 0 ]; then
    echo "Ошибка при запросе данных к API."
    exit 1
fi

# Обработка данных в зависимости от категории
if [ "$CATEGORY" == "people" ]; then
    if [ "$RESPONSE" != "null" ]; then
        echo "Информация о персонаже:"
        echo "Имя: $(echo $RESPONSE | jq -r '.name')"
        echo "Рост: $(echo $RESPONSE | jq -r '.height') см"
        echo "Масса: $(echo $RESPONSE | jq -r '.mass') кг"
        echo "Цвет волос: $(echo $RESPONSE | jq -r '.hair_color')"
        echo "Цвет кожи: $(echo $RESPONSE | jq -r '.skin_color')"
        echo "Цвет глаз: $(echo $RESPONSE | jq -r '.eye_color')"
        echo "Год рождения: $(echo $RESPONSE | jq -r '.birth_year')"
        echo "Пол: $(echo $RESPONSE | jq -r '.gender')"
    else
        echo "Персонаж с ID $ID не найден."
    fi
elif [ "$CATEGORY" == "planets" ]; then
    if [ "$RESPONSE" != "null" ]; then
        echo "Информация о планете:"
        echo "Название: $(echo $RESPONSE | jq -r '.name')"
        echo "Диаметр: $(echo $RESPONSE | jq -r '.diameter') км"
        echo "Климат: $(echo $RESPONSE | jq -r '.climate')"
        echo "Гравитация: $(echo $RESPONSE | jq -r '.gravity')"
        echo "Поверхность воды: $(echo $RESPONSE | jq -r '.surface_water')%"
        echo "Население: $(echo $RESPONSE | jq -r '.population')"
    else
        echo "Планета с ID $ID не найдена."
    fi
elif [ "$CATEGORY" == "films" ]; then
    if [ "$RESPONSE" != "null" ]; then
        echo "Информация о фильме:"
        echo "Название: $(echo $RESPONSE | jq -r '.title')"
        echo "Эпизод: $(echo $RESPONSE | jq -r '.episode_id')"
        echo "Режиссер: $(echo $RESPONSE | jq -r '.director')"
        echo "Продюсеры: $(echo $RESPONSE | jq -r '.producer')"
        echo "Дата выпуска: $(echo $RESPONSE | jq -r '.release_date')"
    else
        echo "Фильм с ID $ID не найден."
    fi
else
    echo "Неверная категория. Доступные категории: people, planets, films."
    exit 1
fi
```

Проверка

```
11:14:51 x 281 ~ ./sw2.sh 2
Usage: ./sw2.sh <category> <id>
Categories: people, planets, films
11:14:54 x 281 ~ ./sw2.sh film 2
Неверная категория. Доступные категории: people, planets, films.
11:15:10 x 282 ~ ./sw2.sh films 2
Информация о фильме:
Название: The Empire Strikes Back
Эпизод: 5
Режиссер: Irvin Kershner
Продюсеры: Gary Kurtz, Rick McCallum
Дата выпуска: 1980-05-17
```

- 4.*** Добавить возможность получать прогноз по выбранному городу на 4 и на 16 дней. Также добавить входящий параметр для выбора единицы измерения температуры (С либо F). Также добавить возможность выбора языка