

Warsztaty z testowania backendu za pomocą narzędzi SoapUI, Postman.

## SoapUI – Odpowiedzi

### Legenda

**PPM** – prawy przycisk myszki, **LPM** – lewy przycisk myszki

### Zadanie S\_01 - Pobierz listę ras

1. Kliknij na "Norad Test Suite"
2. PPM -> New Test Case
3. Podaj nazwę "Exercise S\_01 – Get all races"
4. Kliknij PPM na "Test Steps (0)" -> Add step -> REST Request
5. Wprowadź nazwę "Get all races"
6. Wybierz metodę restową, która będzie pobierać wszystkie rasy  
**/dictionaries/races -> getRacesUsingGET -> Request 1**
7. Zadanie wykonane.
8. Sprawdź działanie klikając w zielony trójkącik.
9. Sprawdź wyniki w prawej części okienka – boczna zakładka "JSON"

### Podpunkt A)

- A.1) Kliknij w zielony plus aby dodać asercję
  - A.2) Wybierz "Compliance, Status and Standards" -> "Valid HTTP Status Codes"
  - A.3) Kliknij w przycisk "Add"
  - A.4) Wpisz 200
- 200 to http code określający poprawne wykonanie requesta po stronie backendu.

**Hint:** <https://jsonpath.com/> - strona wspierająca zrozumienie działania JsonPath

### Podpunkt B )

- B.1) Kliknij w zielony plus aby dodać asercję
- B.2) Wybierz "Property Content" -> "JsonPath Count"
- B.3) W polu "JSONPath Expression" wpisz \$
- B.4) W polu "Expected Result" wprowadź wartość oczekiwaną **10**

Hint: "Select from current" - pobiera wartość z ostatniego wyniku wywołania requesta.

### Podpunkt C)

- C.1) Kliknij w zielony plus aby dodać asercję
  - C.2) Wybierz "Property Content" -> "JsonPath Match"
  - C.3) W polu "JSONPath Expression" wpisz **\$(4).name**
  - C.4) W polu "Expected Result" wprowadź wartość oczekiwaną **Kot egzotyczny**
- Hint: "Select from current" - pobiera wartość z ostatniego wyniku wywołania requesta.

### Zadanie S\_02 - Zarejestruj nowego użytkownika

1. Kliknij na "Norad Test Suite"
2. PPM -> New Test Case
3. Podaj nazwę "Exercise S\_02 – Register new user"
4. Kliknij PPM na "Test Steps (0)" -> Add step -> REST Request
5. Wprowadź nazwę "Register new user"
6. Wybierz metodę restową, która tworzy nowego użytkownika  
  
"/api/accounts->registerUsingPOST->Request 1"
7. Określ zawartość POSTa zgodnie z dokumentacją w Swaggerze dla endpointu, który wykonuje utworzenie nowego użytkownika: <https://norad-beta.duckdns.org/norad/swagger-ui.html#/account-endpoint/registerUsingPOST>

#### Krok 1.

1. PPM na "Exercise S\_02 – Register new User -> Test Steps (1)"
2. Wybierz "Add Step -> Groovy Script", określ nazwę "Generate unique email"
3. Upewnij się, że skrypt "Generate unique email" jest wykonany jako pierwszy
4. Umieść zawartość skryptu

**/postman-soapui-training/exercises-answers/soapui/ex-s02-step-01-generate-random-email.groovy**

Zamiast "mszpiler" wprowadź swoją pierwszą listę imienia i nazwisko.

5. Zmień zawartość contentu wysyłanego w requeście, pole email w JSONIE zmień na wartość

**#{#TestCase#email}**

Przykład znajdziesz w pliku:

**/postman-soapui-training/exercises-answers/soapui/json/register-new-user.json**

#### Krok 2.

1. Kliknij w zielony plusik
2. Wybierz "Property Content" -> "JsonPath Match"
3. W "JSONPath expression" wprowadź **\$.email**
4. W "Expected result" wprowadź **\${#TestCase#email}**

B.5) Żeby sprawdzić czy asercja działa, to dwukrotnie kliknij w nazwę test case **"Exercise S\_02 – Register new user"**, kliknij w zielony trójkącik, jeżeli w dolnym oknie żaden z kroków nie zapalił lampki na czerwono to test przebiegł pozytywnie.

#### Krok 3.

1. Kliknij w zielony plusik
2. Wybierz "Script" -> "Script Assertion"
3. Ustal nazwę asercji na "Database verification"
4. Umieść zawartość skryptu

**/postman-soapui-training/exercises-answers/soapui/ex-s02-step-03-check-database.groovy**

5. Kliknij OK

#### Krok 4.

1. PPM na "Exercise S\_02 – Register new User -> Test Steps (1)"
2. Wybierz "Add Step -> Groovy Script", określ nazwę "Save user UUID in properties"
3. Umieść zawartość skryptu

**/postman-soapui-training/exercises-answers/soapui/ex-s02-step-04-save-user-uuid-in-properties.groovy**

### Zadanie S\_03 - Dodaj nowe zwierzę do nowego użytkownika

1. Kliknij na "Norad Test Suite"
2. PPM -> New Test Case
3. Podaj nazwę "Exercise S\_03 - Add new animal to new user"

#### Krok 1.

1. Kliknij PPM na "Test Steps (0)" -> Add step -> Run TestCase
2. Podaj nazwę "Run TestCase Register new user"
3. W "Target TestCase" wybierz "Exercise S\_02 – Register new user"
4. Kliknij OK.

#### Krok 2.

1. Kliknij PPM na "Test Steps (1)" -> "Add Step" -> "Groovy Script"
2. Podaj nazwę "Generate random animal name"
3. Umieść zawartość pliku:

**/postman-soapui-training/exercises-answers/soapui/ex-S03-step-02-random-animal-name.groovy**

Krok 3.

1. Kliknij PPM na "Test Steps (2)" -> "Add Step" -> "REST Request"
2. Podaj nazwę "Get races"
3. Wybierz request

**/dictionaries/races/ -> getRacesUsingGET -> Request 1**

4. Kliknij PPM na "Test Steps (3)" -> "Add Step" -> "Groovy Script"
5. Podaj nazwę "Fetch random race"
6. Umieść zawartość pliku:

**/postman-soapui-training/exercises-answers/soapui/ex-s03-step-03-fetch-random-race.groovy**

Krok 4.

1. Kliknij PPM na "Test Steps (4)" -> "Add Step" -> "Groovy Script"
2. Podaj nazwę "Get Access Token"
7. Umieść zawartość pliku:

**/postman-soapui-training/exercises-answers/soapui/ex-s03-step-04-get-access-token.groovy**

Krok 5.

1. Kliknij PPM na "Test Steps (5)" -> "Add Step" -> "REST Request".
2. Podaj nazwę "Create new animal".
3. Wybierz request:

**/animals -> createUsingPOST -> Request 1**

4. W lewym dolnym oknie należy wkleić zawartość body requesta, przykład body znajduje się dokumentacji Swagger - <https://norad-beta.duckdns.org/norad/swagger-ui.html#/animal-endpoint/createUsingPOST>. Zweryfikuj ilość tworzonych zdjęć - przykład w Swaggere tworzy tylko jedno zdjęcie. Dodaj jeszcze 2 pozycje na liście "images".
5. Body jest tylko przykładem, należy sparametryzować pola **name**, **racelId**, w następujący sposób - odpowiednio **#{#TestCase#animalName}**, **#{#TestCase#racelId}** zmienne te ustalone są w poprzednich krokach aktualnego TestCase'a.
6. Przykład body docelowego znajdziesz w pliku:

**/postman-soapui-training/exercises-answers/soapui/json/create-new-animal.json**

Request "Create new animal" wymaga autoryzacji, zatem trzeba określić w headerze parametr "Authorization", zatem należy wykonać kolejne punkty:

7. Kliknij PPM na krok "Create new animal"
8. Przejdź do zakładki "Headers" i kliknij w zielony krzyżyk aby dodać nowy parametr do headera.
9. Nazwa parametru **Authorization**, wartość parametru:

**Bearer \${#TestCase#ACCESS\_TOKEN}**

Property ACCESS\_TOKEN pobrane było we wcześniejszym kroku o nazwie "Get Access Token"

Krok 6.

1. Kliknij PPM na "Test Steps (6)" -> "Add Step" -> "Groovy Script"
2. Podaj nazwę "Save animal UUID in test suite properties".
3. Umieść zawartość skryptu

**/postman-soapui-training/exercises-answers/soapui/ex-s03-step-06-save-animal-uuid-in-test-suite-properties.groovy**

Krok 7.

1. Kliknij PPM na "Test Steps (7)" -> "Add Step" -> "Groovy Script"
1. Podaj nazwę "Save first image UUID in test suite properties".
2. Umieść zawartość skryptu

**/postman-soapui-training/exercises-answers/soapui/ex-s03-step-07-save-first-image-uuid-in-test-suite-properties**

Krok 8.

3. Kliknij PPM na "Test Steps (8)" -> "Add Step" -> "Groovy Script"
2. Podaj nazwę "Save user email in test suite properties".
4. Umieść zawartość skryptu

**/postman-soapui-training/exercises-answers/soapui/ex-s03-step-08-save-user-email-in-test-suite-properties**

Asercja A

1. Kliknij dwukrotnie na step "Create new animal".
2. Kliknij w zielony krzyżyk aby dodać asercję.
3. Wybierz "Script" -> "Script Assertion"
4. Zmień nazwę asercji na "Assert A – expected race"
5. Umieść zawartość pliku:

**/postman-soapui-training/exercises-answers/soapui/ex-s03-assert-A-expected-race.groovy**

Asercja B

1. Kliknij w zielony krzyżyk aby dodać asercję.
2. Wybierz "Script" -> "Script Assertion"
3. Zmień nazwę asercji na "Assert B – expected user"
4. Umieść zawartość pliku:

**/postman-soapui-training/exercises-answers/soapui/ex-s03-assert-B-expected-user.groovy**

Asercja C

1. Kliknij w zielony krzyżyk aby dodać asercję.
2. Wybierz "Property Content" -> "JsonPath Count"

1. Zmień nazwę asercji na "Assert C - expected images count"
3. W okienku "JSONPath Expression" wprowadź tekst: **\$.images**
4. W okienku "Expected result" wprowadź wartość: **3**
5. Kliknij w tekst "Test" aby sprawdzić czy

#### Asercja D

1. Kliknij w zielony krzyżyk aby dodać asercję.
2. Wybierz "Script" -> "Script Assertion"
3. Zmień nazwę asercji na "Assert D – uuids exists in images"
4. Umieść zawartość pliku:

**/postman-soapui-training/exercises-answers/soapui/ex-s03-assert-D-uuid-exists-in-images.groovy**

#### Zadanie S\_04 – Wykonaj upload zdjęcia - załącznik wskazany ręcznie

1. Kliknij na "Norad Test Suite"
2. PPM -> New Test Case
3. Podaj nazwę "Exercise 7 – Upload file"

#### Krok 1.

1. Kliknij PPM na "Test Steps (2)" -> "Add Step" -> "Groovy Script"
2. Podaj nazwę "Get Access Token"
3. Umieść zawartość pliku:

**/postman-soapui-training/exercises-answers/soapui/ex-s04-step-02-get-access-token.groovy**

#### Krok 2.

1. Kliknij PPM na "Test Steps (3)" -> "Add Step" -> "REST Request"
2. Podaj nazwę "Upload file"
3. Wybierz request:

**/animal/{animalUUID}/images/{imageUUID} -> uploadImageUsingPOST -> Request 1**

Request "Upload file" wymaga autoryzacji, zatem trzeba określić w headerze parametr "Authorization", zatem należy wykonać kolejne punkty:

4. Kliknij PPM na krok "Create new animal"
5. Przejdź do zakładki "Headers" i kliknij w zielony krzyżyk aby dodać nowy parametr do headera.
6. Nazwa parametru **Authorization**, wartość parametru:

**Bearer \${#TestCase#ACCESS\_TOKEN}**

Property ACCESS\_TOKEN pobrane było we wcześniejszym kroku o nazwie "Get Access Token".

### Krok 3.

1. Kliknij w zakładkę **“Attachments”**, pojawi się okienko z listą załączników, na początku lista jest pusta.
2. Kliknij w zielony krzyżyk w okienku z załącznikami.
3. Wybierz plik ze zdjęciem dostępny na dysku.
4. W górnym okienku zawierającym listę parametrów **animalUUID**, **imageUUID** oraz **file** należy określić ich zawartość. Parametr **file** przyjmuje wartość **file:<nazwa\_pliku>** gdzie **<nazwa\_pliku>** to nazwa pliku ze zdjęciem, które wybrałeś na swoim dysku.
5. Dla parametru **animalUUID** wprowadź wartość **#{#TestSuite#animalUUID}**
6. Dla parametru **imageUUID** wprowadź wartość **#{#TestSuite#imageUUID}**

### Asercja A

1. Na zaznaczonym stepie **“Upload file”** kliknij w zielony krzyżyk aby dodać asercję.
2. Wybierz **“Compliance, Status and Standards”** -> **“Valid Http Status Codes”**
3. Wprowadź wartość **200**

### Asercja B

1. Kliknij PPM na **“Test Steps (3)”** -> **“Add Step”** -> **“REST Request”**
2. Podaj nazwę **“Download file”**
3. Wybierz request:

**/animal/{animalUUID}/images/{imageUUID} -> downloadImageUsingGET -> Request 1**

Request **“Download file”** wymaga autoryzacji, zatem trzeba określić w headerze parametr **“Authorization”**, zatem należy wykonać kolejne punkty:

4. Kliknij PPM na krok **“Create new animal”**
5. Przejdź do zakładki **“Headers”** i kliknij w zielony krzyżyk aby dodać nowy parametr do headera.
6. Nazwa parametru **Authorization**, wartość parametru:

**Bearer #{#TestCase#ACCESS\_TOKEN}**

Property **ACCESS\_TOKEN** pobrane było we wcześniejszym kroku o nazwie **“Get Access Token”**.

7. Dla parametru **animalUUID** wprowadź wartość **#{#TestSuite#animalUUID}**
8. Dla parametru **imageUUID** wprowadź wartość **#{#TestSuite#imageUUID}**
9. W okienku kroku **“Download file”** kliknij w zielony krzyżyk aby dodać asercję.
10. Wybierz **“Script”** -> **“Script Assertion”**
11. Umieść zawartość pliku:

**/postman-soapui-training/exercises-answers/soapui/ex-s04-assert-B-validate-filesize.groovy**

12. Na liście asercji zmień nazwę nowo utworzonej asercji na **“Validate file size”**.

Zadanie S\_05 – Wykonaj upload zdjęcia - załącznik wskazany za pomocą skryptu Groovy

1. Kliknij na "Norad Test Suite"
2. PPM -> New Test Case
3. Podaj nazwę "Exercise S\_05 – Upload file - Groovy"

#### Krok 1.

4. Kliknij PPM na "Test Steps (2)" -> "Add Step" -> "Groovy Script"
5. Podaj nazwę "Get Access Token"
6. Umieść zawartość pliku:

**/postman-soapui-training/exercises-answers/soapui/ex-s04-step-02-get-access-token.groovy**

#### Krok 2.

1. Kliknij PPM na "Test Steps (0)" -> "Add Step" -> "REST Request"
2. Podaj nazwę "Upload file"
3. Wybierz request:

**/animal/{animalUUID}/images/{imageUUID} -> uploadImageUsingPOST -> Request 1**

Request "Upload file" wymaga autoryzacji, zatem trzeba określić w headerze parametr "Authorization", zatem należy wykonać kolejne punkty:

4. Kliknij PPM na krok "Create new animal"
5. Przejdź do zakładki "Headers" i kliknij w zielony krzyżyk aby dodać nowy parametr do headera.
6. Nazwa parametru **Authorization**, wartość parametru:

**Bearer \${#TestCase#ACCESS\_TOKEN}**

Property ACCESS\_TOKEN pobrane było we wcześniejszym kroku o nazwie "Get Access Token".

7. Dla parametru animalUUID wprowadź wartość **\${#TestSuite#animalUUID}**
8. Dla parametru imageUUID wprowadź wartość **\${#TestSuite#imageUUID}**
9. Dla parametru file wprowadź wartość: **file:\${#Project#imageFileName}**

#### Krok 3.

1. Kliknij na nazwę **projektu** w SoapUI.
2. Przejdź na zakładkę "Custom Properties"
3. Za pomocą zielonego krzyżyka dodaj dwa properties:
  - o **imageFilePath**
  - o **imageFileName**
4. Wartości to ścieżka i nazwa do pliku ze zdjęciem na Twoim komputerze.

#### Krok 4.

1. Kliknij PPM na "Test Steps (2)" -> "Add Step" -> "Groovy Script"
2. Podaj nazwę "Setup attachment"
3. Umieść zawartość skryptu:

**/postman-soapui-training/exercises-answers/soapui/ex-s05-step-04-setup-attachment.groovy**



#### Asercja A

1. Na zaznaczonym stepie "Upload file" kliknij w zielony krzyżyk aby dodać asercję.
2. Wybierz "Compliance, Status and Standards" -> "Valid Http Status Codes"
3. Wprowadź wartość **200**

#### Asercja B

1. Kliknij PPM na "Test Steps (4)" -> "Add Step" -> "REST Request"
2. Podaj nazwę "Download file"
3. Wybierz request:

**/animal/{animalUUID}/images/{imageUUID} -> downloadImageUsingGET -> Request 1**

Request "Download file" wymaga autoryzacji, zatem trzeba określić w headerze parametr "Authorization", zatem należy wykonać kolejne punkty:

4. Kliknij PPM na krok "Create new animal"
5. Przejdź do zakładki "Headers" i kliknij w zielony krzyżyk aby dodać nowy parametr do headera.
6. Nazwa parametru **Authorization**, wartość parametru:

**Bearer \${#TestCase#ACCESS\_TOKEN}**

7. Dla parametru animalUUID wprowadź wartość **\${#TestSuite#animalUUID}**
8. Dla parametru imageUUID wprowadź wartość **\${#TestSuite#imageUUID}**
9. W okienku kroku "Download file" kliknij w zielony krzyżyk aby dodać asercję.
10. Wybierz "Script" -> "Script Assertion"
11. Umieść zawartość pliku:

**/postman-soapui-training/exercises-answers/soapui/ex-s05-assert-B-validate-filesize.groovy**

12. Na liście asercji zmień nazwę nowo utworzonej asercji na "Validate file size".

#### Zadanie S\_06 Dodaj zwierzęta pobierając dane z pliku CSV

1. Kliknij na "Norad Test Suite"
2. PPM -> New Test Case
3. Podaj nazwę "Exercise S\_06 – Add many animals from CSV file"

##### Krok 1.

1. Kliknij PPM na "Test Steps (0)" -> "Add Step" -> "Run TestCase"
2. Podaj nazwę: "Run TestCase Register new user"
3. Wybierz z listy TestCase: "Exercise S\_03 – Register new user"
4. Zaznacz "email" w sekcji "Return Properties".

##### Krok 2.

1. Kliknij PPM na "Test Steps (2)" -> "Add Step" -> "Groovy Script"
2. Podaj nazwę "Get Access Token"
3. Umieść zawartość pliku:

</postman-soapui-training/exercises-answers/soapui/ex-s03-step-04-get-access-token.groovy>

Krok 3.

1. Kliknij PPM na "Test Steps (1)" -> "Add Step" -> "REST Request"
2. Podaj nazwę: "Create one animal".
3. Wybierz request:

***/animals -> createUsingPOST -> Request 1***

4. W lewym dolnym okienku należy określić zawartość body requesta tworzącego zwierzę
5. Przykład body znajdziesz w pliku:

***/postman-soapui-training/exercises-answers/soapui/json/create-new-animal.json***

6. Należy step "Create one animal" ustawić na **Disabled** ażeby domyślnie step nie był wykonywany przez SoapUI tylko na wyraźne wywołanie w skrypcie Groovy.

Request "Create one animal" wymaga autoryzacji, zatem trzeba określić w headerze parametr "Authorization", zatem należy wykonać kolejne punkty:

7. Kliknij PPM na krok "Create new animal"
8. Przejdź do zakładki "Headers" i kliknij w zielony krzyżyk aby dodać nowy parametr do headera.
9. Nazwa parametru **Authorization**, wartość parametru:

**Bearer \${#TestCase#ACCESS\_TOKEN}**

Krok 4.

1. Kliknij PPM na "Test Steps (1)" -> "Add Step" -> "Groovy Script"
2. Podaj nazwę: "Loop CSV file"
3. Umieść zawartość skryptu:

***/postman-soapui-training/exercises-answers/soapui/ex-s06-step-04-loop-csv-file.groovy***

Asercja B:

1. Kliknij PPM na "Test Steps (1)" -> "Add Step" -> "Groovy Script"
2. Podaj nazwę: "Validate animals in database"
3. Umieść zawartość skryptu:

***/postman-soapui-training/exercises-answers/soapui/ex-s06-assert-B-validate-animals-in-database.groovy***

Zadanie S\_07 Wiele kroków

1. Kliknij na "Norad Test Suite"

2. PPM -> New Test Case
3. Podaj nazwę "Exercise S\_07 – Many steps"

Krok 1.

1. Kliknij PPM na "Test Steps (0)" -> "Add Step" -> "Run TestCase"
2. Podaj nazwę: "Run TestCase Register new user"
3. Wybierz z listy TestCase: "Exercise S\_02 – Register new user"

Krok 2.

1. Kliknij PPM na "Test Steps (2)" -> "Add Step" -> "Groovy Script"
2. Podaj nazwę "Get Access Token"
3. Umieść zawartość pliku:

***/postman-soapui-training/exercises-answers/soapui/ex-s03-step-04-get-access-token.groovy***

Krok 3.

1. Kliknij PPM na "Test Steps (1)" -> "Add Step" -> "REST Request"
2. Podaj nazwę: "Create one animal".
3. Wybierz request:

***/animals -> createUsingPOST -> Request 1***

4. W lewym dolnym okienku należy określić zawartość body requesta tworzącego zwierzę
5. Przykład body znajdziesz w pliku:

***/postman-soapui-training/exercises-answers/soapui/json/create-new-animal.json***

6. Należy step "Create one animal" ustawić na **Disabled** ażeby domyślnie step nie był wykonywany przez SoapUI tylko na wyraźne wywołanie w skrypcie Groovy.

Request "Create one animal" wymaga autoryzacji, zatem trzeba określić w headerze parametr "Authorization", zatem należy wykonać kolejne punkty:

7. Kliknij PPM na krok "Create new animal"
8. Przejdź do zakładki "Headers" i kliknij w zielony krzyżyk aby dodać nowy parametr do headera.
9. Nazwa parametru **Authorization**, wartość parametru:

**Bearer \${#TestCase#ACCESS\_TOKEN}**

Krok 4.

1. Kliknij PPM na "Test Steps (2)" -> "Add Step" -> "Groovy Script"
2. Podaj nazwę: "Loop CSV file"
3. Umieść zawartość skryptu:

***/postman-soapui-training/exercises-answers/soapui/ex-s07-step-04-loop-csv-file.groovy***

Krok 5.

1. Kliknij PPM na "Test Steps (3)" -> "Add Step" -> "REST Request"

2. Podaj nazwę "Get animal details"
3. Wybierz request:

**/animals/{animalUUID} -> getAnimalUsingGET-> Request 1**

4. Ustaw na **"Disabled"** krok **"Get animal details"** - krok będzie uruchamiany na wyraźne wywołanie w skrypcie Groovy.
5. Ustawienie wykonujemy PPM na stepie "Get animal details" -> "Disable TestStep"

Request "Get animal details" wymaga autoryzacji, zatem trzeba określić w headerze parametr "Authorization", zatem należy wykonać kolejne punkty:

6. Kliknij PPM na krok "Create new animal"
7. Przejdź do zakładki "Headers" i kliknij w zielony krzyżyk aby dodać nowy parametr do headera.
8. Nazwa parametru **Authorization**, wartość parametru:

**Bearer \${#TestCase#ACCESS\_TOKEN}**

9. Kliknij PPM na "Test Steps (4)" -> "Add Step" -> "Groovy Script"
10. Podaj nazwę: "Validate animals using REST API"
11. Umieść zawartość skryptu:

**/postman-soapui-training/exercises-answers/soapui/ex-s07-step-05-validate-animals-using-rest-api.groovy**