

# JMeter Warsztat



# UWAGI

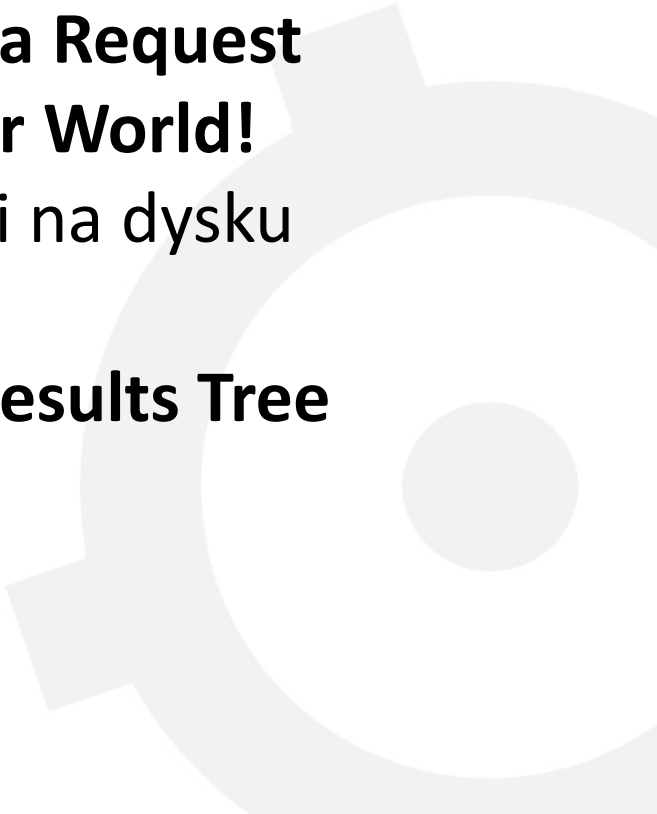
Każde ćwiczenie powinno być zapisane jako pojedynczy plan testów – osobny plik.

Ćwiczenia (niektóre) tworzą integralną całość – niewykonanie ćwiczenia może skutkować problemami lub brakiem możliwości realizacji następnego ćwiczenia.

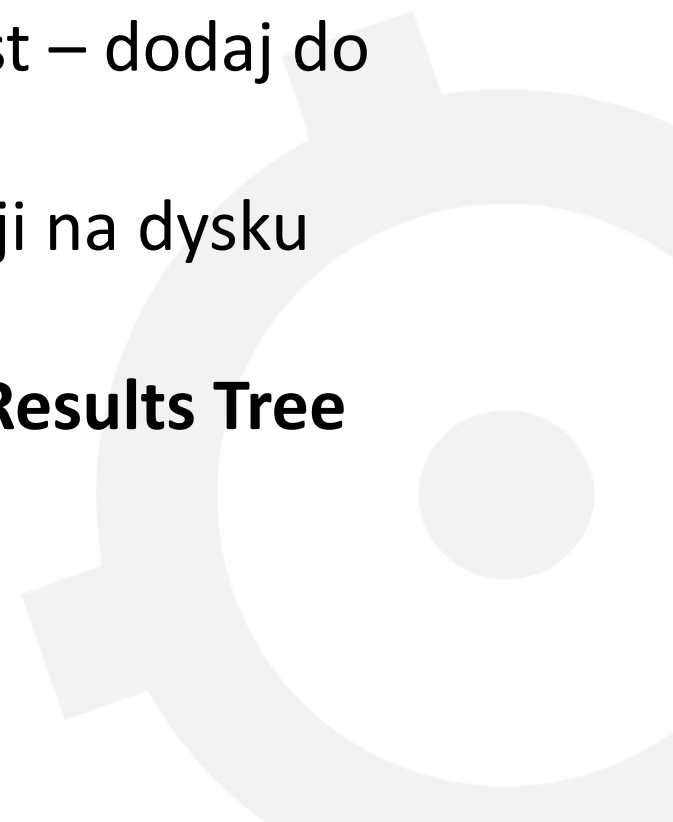
Zrealizowane ćwiczenia – przygotowane pliki są podstawą do zaliczenia zadania.

Zrealizowanie ćwiczeń jest obowiązkowe.

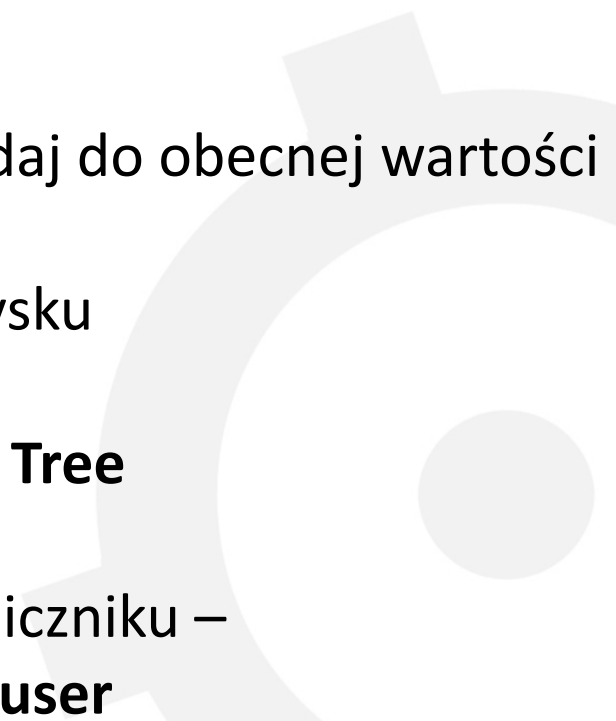
# ĆWICZENIE 1.1

1. Włącz Apache JMeter™
  2. Do Test Planu dodaj **View Results Tree**
  3. Do Test Planu dodaj **Thread Group**
  4. Do Thread Group dodaj sampler – **Java Request**
  5. W polu **Label** dodaj tekst **Hello JMeter World!**
  6. Zapisz **Test Plan** w wybranej lokalizacji na dysku
  7. Wykonaj **Test Plan**
  8. Sprawdź szczegółowy wynik w **View Results Tree**
- 

# ĆWICZENIE 1.2

1. W ramach Thread Group ustaw:  
**Number of Threads**, np. 10  
**Loop Count**, np. 2
  2. W polu **Label** samplera – Java Request – dodaj do obecnej wartości **\${\_\_threadNum}**.
  3. Zapisz **Test Plan** w wybranej lokalizacji na dysku
  4. Wykonaj Test Plan
  5. Sprawdź szczegółowy wynik w **View Results Tree**
- 

# ĆWICZENIE 1.3

1. Do Therad Group dodaj **Counter** (Config Elements)
  2. W liczniku ustaw wartość początkową – **Start** – 1
  3. W liczniku ustaw wartość przyrostu – **Increment** – 1
  4. W liczniku ustaw nazwę referencyjną –  
**Exported Variable Name** – count
  5. W polu **Label** samplera – Java Request – dodaj do obecnej wartości odwołanie do licznika **`${count}`**
  6. Zapisz **Test Plan** w wybranej lokalizacji na dysku
  7. Wykonaj Test Plan
  8. Sprawdź szczegółowy wynik w **View Results Tree**
  9. Powtórz to ćwiczenie z zaznaczoną opcją w liczniku –  
**Track counter independently for each user**
- 

# ĆWICZENIE 1.4

1. W polu **Label** samplera – Java Request – dodaj do obecnej wartości odwołanie do funkcji

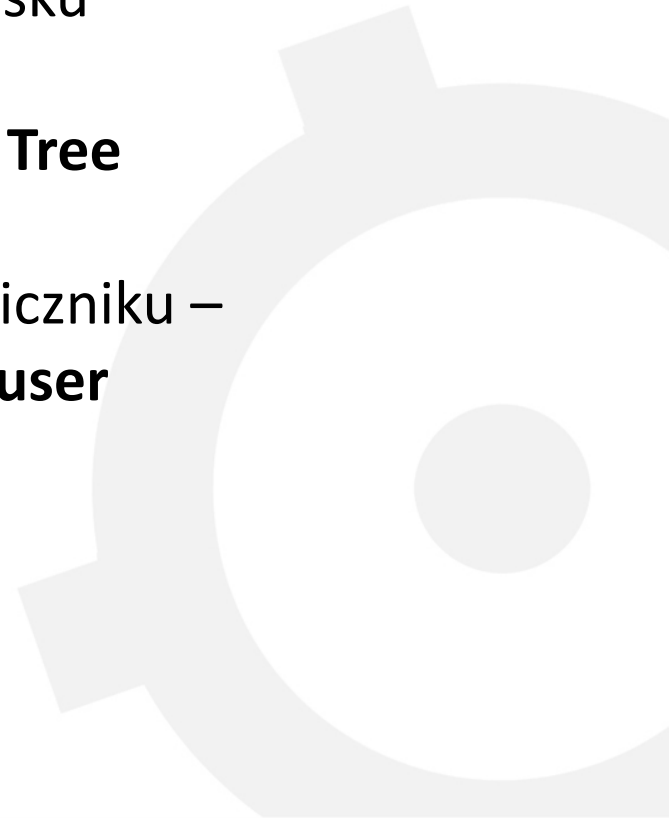
**`${__RandomString(${count},abcdefghijklmnopqrstuvwxyz)}`**

2. Zapisz **Test Plan** w wybranej lokalizacji na dysku

3. Wykonaj Test Plan

4. Sprawdź szczegółowy wynik w **View Results Tree**

5. Powtórz to ćwiczenie z zaznaczoną opcją w liczniku –  
**Track counter independently for each user**



# ĆWICZENIE 1.5

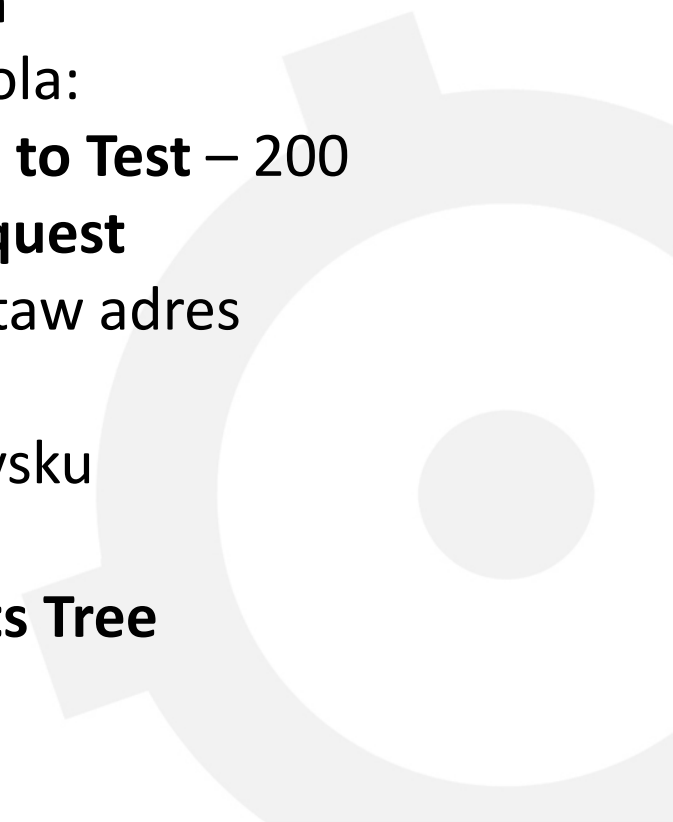
1. Do Test Planu dodaj **User Defined Variables** (Config Elements)
2. Dodajemy zmienną **timestamp** i wartość **`${__time()}`**
3. Do Test Planu dodaj grupę wątków – **SetUp (Threads)**
4. W grupie wątków – **SetUp** – dodaj **Java Request** z **Label** ustawionym na **Tests started at: `${__time(HH:mm:ss)}`**
5. Do Test Planu dodaj grupę wątków – **TearDown (Threads)**
6. W grupie wątków – **TearDown** – dodaj **Java Request** z **Label** ustawionym na **Tests finished at: `${__time(HH:mm:ss)}`**
7. W grupie wątków – **TearDown** – dodaj **Java Request** z **Label** ustawionym na **Total testing time: `${__longSum($__time()),-${timestamp}}` ms.**
8. Zapisz **Test Plan** w wybranej lokalizacji na dysku
9. Wykonaj Test Plan
10. Sprawdź szczegółowy wynik w **View Results Tree**

# ĆWICZENIE 1.6

1. W polu **ResponseCode** głównego samplera – Java Request – ustaw wartość, np. 200
2. Do samplera – Java Request – dodaj **Response Assertion**
3. Ustaw w **Response Assertion** następujące pola:  
**Response Code, Equals** i dodaj **Pattern to Test** – 200
4. Zapisz **Test Plan** w wybranej lokalizacji na dysku
5. Wykonaj Test Plan
6. Sprawdź szczegółowy wynik w **View Results Tree**
7. Powtórz to ćwiczenie dla **ResponseCode** – 404 lub 500



# ĆWICZENIE 2.1

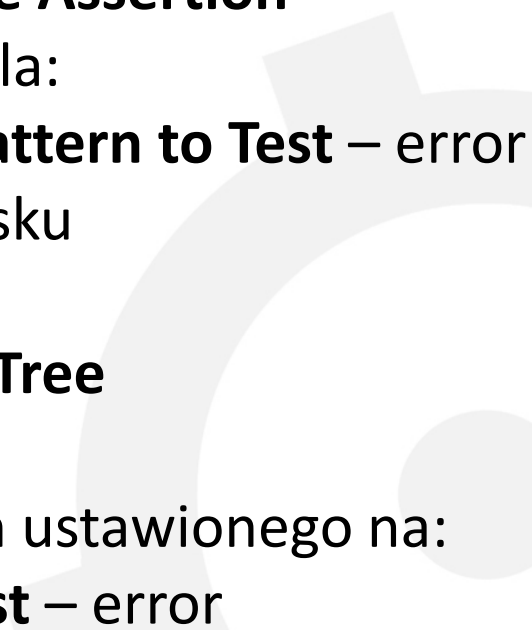
1. Włącz Apache JMeter™
  2. Do Test Planu dodaj **View Results Tree**
  3. Do Test Planu dodaj **Thread Group**
  4. Do Thread Group dodaj **Response Assertion**
  5. Ustaw w Response Assertion następujące pola:  
**Response Code, Equals** i dodaj **Pattern to Test – 200**
  6. Do Thread Group dodaj sampler – **HTTP Request**
  7. W polu **Path** samplera – **HTTP Request** – ustaw adres  
**http://google.com**
  8. Zapisz **Test Plan** w wybranej lokalizacji na dysku
  9. Wykonaj Test Plan
  10. Sprawdź szczegółowy wynik w **View Results Tree**
- 

# ĆWICZENIE 2.2

1. Do samplera – HTTP Request dodaj **Response Assertion**
2. Ustaw w Response Assertion następujące pola:  
**Text Response, Contains** i dodaj **Pattern to Test** – Google
3. Zapisz **Test Plan** w wybranej lokalizacji na dysku
4. Wykonaj Test Plan
5. Sprawdź szczegółowy wynik w **View Results Tree**



# ĆWICZENIE 2.3

1. Do Thread Group dodaj sampler – **HTTP Request**
  2. W polu **Path** samplera – HTTP Request – wstaw adres  
**`http://google.com/1234`**
  3. Do samplera – HTTP Request dodaj **Response Assertion**
  4. Ustaw w Response Assertion następujące pola:  
**Text Response, Contains, Not** i dodaj **Pattern to Test** – error
  5. Zapisz **Test Plan** w wybranej lokalizacji na dysku
  6. Wykonaj Test Plan
  7. Sprawdź szczegółowy wynik w **View Results Tree**
  8. Powtórz to ćwiczenie dla Response Assertion ustawionego na:  
**Text Response, Contains, Pattern to Test** – error
- 

# ĆWICZENIE 2.4

1. Do Thread Group dodaj element konfiguracyjny  
**HTTP Request Defaults**
2. W polu **Server Name or IP** elementu konfiguracyjnego – **HTTP Request Defaults** – wstaw adres np. **google.com**
3. We wszystkich samplerach – **HTTP Request** - zmień wartość pól **Path** na:  

<b>http://google.com/</b>	<b>→</b>	<b>/</b>
<b>http://google.com/1234</b>	<b>→</b>	<b>/1234</b>
4. Zapisz **Test Plan** w wybranej lokalizacji na dysku
5. Wykonaj Test Plan
6. Sprawdź szczegółowy wynik w **View Results Tree**

# ĆWICZENIE 3 [DODATKOWE] 1/3

1. Włącz Apache **JMeter™**
2. Do WorkBench dodaj **HTTP(S) Test Script Recorder**
3. W **HTTP(S) Test Script Recorder** ustaw port, np. **8888**
4. W **HTTP(S) Test Script Recorder** ustaw **Target Controller** na **WorkBench -> HTTP(S) Test Script Recorder**
5. Skonfiguruj przeglądarkę do komunikacji przez serwer proxy **localhost / 8888**
6. W **HTTP(S) Test Script Recorder** naciśnij **START**
7. Zarejestruj proces logowania do wersji demo aplikacji **TestArena**
8. W **HTTP(S) Test Script Recorder** naciśnij **STOP**
9. Do **Test Planu** dodaj **View Results Tree**
10. Do **Test Planu** dodaj **Thread Group**
11. Do **Thread Group** dodaj **Transaction Controller** (Logic Controller)

# ĆWICZENIE 3 [DODATKOWE] 2/3

12. W polu **Name** elementu **Transaction Controller** – dodaj przykładowy nazwę ( **LOGOWANIE** ).

16. Przenieś zarejestrowane elementy do

**Transaction Controller ( LOGOWANIE )**

17. Do zapytania związanego ze stroną logowania dodaj **Regular Expression Extractor** (Post Processors)

18. W **Regular Expression Extractor** uzupełnij pola –

**Reference Name:** csrf

**Regular Expression:** name="csrf" value="([0-9a-z]+)"

**Template:** \$1\$

**Match No.:** 1

19. W zapytaniu związanym z logowaniem zmień wartość parametru **csrf** na **\${csrf}**

# ĆWICZENIE 3 [DODATKOWE] 3/3

- 20. Do **Thread Group** dodaj **HTTP Cookie Manager**
- 21. Do **Thread Group** dodaj **HTTP Cache Manager**
- 22. Zapisz **Test Plan** w wybranej lokalizacji na dysku
- 23. Wykonaj **Test Plan**
- 24. Sprawdź szczegółowy wynik w **View Results Tree**



# ĆWICZENIE 4 [DODATKOWE]

**Przygotuj skrypt logujący się do dowolnej (innej) aplikacji**

