

LAB8: JAVA teoria:

1. Format JSON – format zapisu obiektu, tablicy, tablicy obiektów

JSON jest prostym formatem wymiany danych., powstał w oparciu o dwie struktury:

- Zbiór par nazwa/wartość. W różnych językach jest to implementowane jako *obiekt*, rekord, struktura, słownik, tabela hash, lista z kluczem, albo tabela asocjacyjna.
- Uporządkowana lista wartości. W większości języków implementuje się to za pomocą *tabeli*, wektora, listy, lub sekwencji.

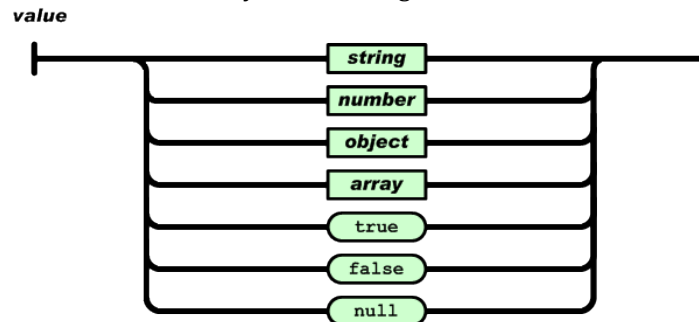
Obiekt jest nieuporządkowanym zbiorem par nazwa/wartość. Opis obiektu:

obiekt { nazwa1: wartość1, nazwa2: wartość2, ... }

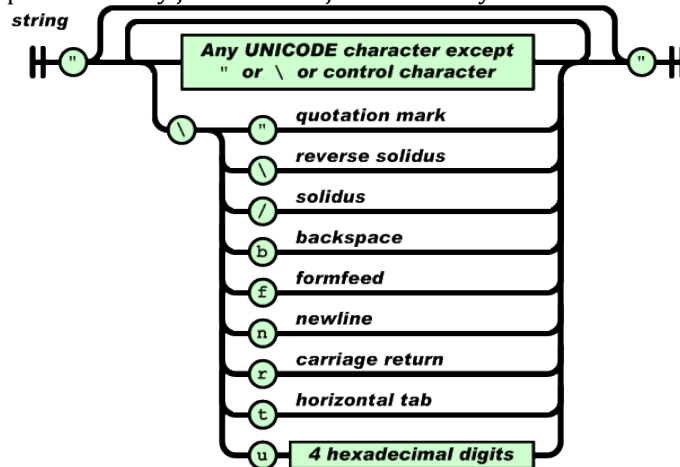
Tablica jest uporządkowanym zbiorem wartości. Opis tablicy:

array [wartość1, wartość2, wartość3, ...]

Łańcuch znakowy, którego początek i koniec oznacza podwójny cudzysłów, lub *liczba*, lub wartość true lub false lub null lub *obiekt* lub *tabela*. Struktury te można zagnieżdżać.



Łańcuch znakowy jest zbiorem zera lub większej ilości znaków Unicode, opakowanym w podwójne cudzysłowy, stosujących znak odwrotnego ukośnika jako początek sekwencji specjalnej (escape). Pojedynczy znak jest reprezentowany jako łańcuch jednoznakowy.



2. Czym są aplikacje REST

REST - to po prostu architektura Internetu. Podstawowe założenie tej architektury brzmi następująco:

Każdy zasób posiada unikalny identyfikator (URI).

Czyli de facto jest to zasada rządząca Siecią - każda podstrona = inny adres. Dodatkowo czynności, jakie chcemy zrobić na danym zasobie są oznaczone przez tzw. HTTP verbs. Dwa najbardziej znane to GET i POST (wykorzystywane choćby w formularzach na stronach internetowych), ale są też inne, takie jak: PUT, DELETE czy OPTIONS.

W przypadku REST mamy adresy URL które są pewnego rodzaju identyfikatorami. Wysyłamy na te adresy zapytanie, które może być zarówno JSONem, XMLem, ale też zwykłym tekstem czy danymi binarnymi. To, co powinno się wydarzyć jest określone przez użytą metodę protokołu HTTP (mamy ich 7 — np. GET pobiera element, DELETE usuwa go itd.).

Pola klas możemy wysyłać w komplecie, a część możemy pominąć (czasami). Odpowiedź może być w formacie JSON, XML, lub też zależnie od tego, jakiego typu odpowiedzi zażyczy sobie klient.

Słowem: mamy dość dużą dowolność w zakresie implementacji, ale jest ona obciążona dość dużą niepewnością.

REST na pewno jest bardzo logiczny jeśli chodzi o operacje na danych — dodawanie, odczyt, aktualizacja itp.

3. Komunikacja HTTP – podstawy, budowa zapytania, budowa odpowiedzi, statusy odpowiedzi (200, 201, 400, 404, 500), kategorie statusów http

Protokół HTTP jest użyteczny, ponieważ udostępnia znormalizowany sposób komunikowania się komputerów ze sobą. Określa on formę żądań klienta (tj. np. przeglądarki www) dotyczących danych oraz formę odpowiedzi serwera na te żądania. Jest zaliczany do protokołów **bezstanowych** (ang. *stateless*) z racji tego, że nie zachowuje żadnych informacji o poprzednich transakcjach z klientem (po zakończeniu transakcji wszystko "przepada"). Pozwala to znacznie zmniejszyć obciążenie serwera, jednak jest kłopotliwe w sytuacji, gdy np. trzeba zapamiętać konkretny stan dla użytkownika, który wcześniej łączył się już z serwerem. Najczęstszym rozwiązaniem tego problemu jest wprowadzenie mechanizmu ciasteczek. Inne podejścia to m.in. sesje po stronie serwera, ukryte parametry (gdy aktualna strona zawiera formularz) oraz parametry umieszczone w URL-u (jak np. /index.php?userid=3).

Konkretne metody protokołu HTTP odpowiadają za odpowiednie operacje:

1. **GET** – pobranie zasobu wskazanego przez URI, może mieć postać warunkową jeśli w nagłówku występują pola warunkowe takie jak "If-Modified-Since"
2. **HEAD** – pobiera informacje o zasobie, stosowane do sprawdzania dostępności zasobu
3. **PUT** – przyjęcie danych przesyłanych od klienta do serwera, najczęściej aby zaktualizować wartość encji
4. **POST** – przyjęcie danych przesyłanych od klienta do serwera
5. **DELETE** – żądanie usunięcia zasobu, włączone dla uprawnionych użytkowników
6. **OPTIONS** – informacje o opcjach i wymaganiach istniejących w kanale komunikacyjnym
7. **TRACE** – diagnostyka, analiza kanału komunikacyjnego
8. **CONNECT** – żądanie przeznaczone dla serwerów pośredniczących pełniących funkcje tunelowania
9. **PATCH** – aktualizacja części danych

Kody powodzenia:

200 OK - Zawartość żadanego dokumentu (najczęściej zwracany nagłówek odpowiedzi).

201 Created - Utworzono – wysłany dokument został zapisany na serwerze.

Kody błędu aplikacji klienta:

- | | | |
|-----|-------------|--|
| 400 | Bad Request | Nieprawidłowe zapytanie – żądanie nie może być obsłużone przez serwer z powodu nieprawidłowości postrzeganej jako błąd użytkownika (np. błędna składnia zapytania) |
| 404 | Not Found | Nie znaleziono – serwer nie odnalazł zasobu według podanego URL ani niczego co by wskazywało na istnienie takiego zasobu w przeszłości |

Kody błędu serwera http

- | | | |
|-----|-----------------------|---|
| 500 | Internal Server Error | Wewnętrzny błąd serwera – serwer napotkał niespodziewane trudności, które uniemożliwiły zrealizowanie żądania |
|-----|-----------------------|---|

Żądanie HTTP

Klient wysyła żądanie do serwera w formie wiadomości. Wiadomość ta ma dokładnie zdefiniowany format:

- linia określająca czasownik HTTP, zasób i wersję protokołu,
- linie zawierające nagłówki,
- pustą linię określającą koniec nagłówków,
- ciało wiadomości (jeśli istnieje).

Odpowiedź HTTP

Serwer odpowiada na żądanie klienta wysyłając odpowiedź⁴. Podobnie jak w przypadku zapytania format jest dokładnie określony:

- linijka z wersją protokołu i statusem odpowiedzi,
- linie zawierające nagłówki,
- pustą linię określającą koniec nagłówków,
- ciało wiadomości (jeśli istnieje).

4. Spring

Spring jest szkieletem tworzenia aplikacji (ang. application framework) w języku Java. Spring powstał jako alternatywa dla programowania aplikacji z użyciem Enterprise JavaBeans.

Programowanie z użyciem EJB narzucało wiele ograniczeń – wymagając między innymi przyjęcia określonego modelu tworzenia oprogramowania. Funkcjonalność EJB okazała się także za "ciężka" do wszystkich zastosowań (w małych projektach wykorzystywano tylko niewielką część oferowanej przez EJB funkcjonalności) jednocześnie stworzenie małej aplikacji w środowisku EJB wymagało nakładu pracy jak przy aplikacji dużej. Odmierna koncepcja Springa – lekkiego szablonu, który nie wymusza specyficznego modelu programowania, stała się bardzo popularna wśród programistów Javy.

Spring oferuje dużą swobodę w tworzeniu rozwiązań, a jednocześnie jest dobrze udokumentowany i zawiera rozwiązania wielu zagadnień, często występujących w programowaniu.

Podczas gdy bazowe komponenty Springa mogą być używane praktycznie w każdej aplikacji, istnieje w nim wiele rozszerzeń, które pozwalają budować aplikacje webowe na bazie Java EE.

5. Springboot vs Spring

Spring Boot jest punktem wyjścia do budowania wszystkich aplikacji opartych na Spring. Spring Boot zaprojektowano tak, aby uruchomić go tak szybko, jak to możliwe, przy minimalnej konfiguracji Spring z góry.

Spring boot pozwala także tworzyć uruchamialne pliki war. - nie potrzebujesz tomcata, ani żadnego innego serwera do odpalenia programu napisanego w Boot. Aplikacje oparte o spring boot możesz wrzucać na serwisy typu heroku. "Zwykły" Spring wymaga za to więcej konfiguracji (XML, lub przy pomocy Adnotacji) i potrzebuje serwera (np. Tomcat, webshpere).

6. Co może zwracać kontroler w Spring Boot? Co oznacza skrót POJO

Plain Old Java Object (POJO) – termin używany przez zwolenników idei mówiącej, że im prostszy design tym lepiej. Używa się go dla określenia obiektów, będących zwyczajnymi obiektami Java, nie zaś obiektami specjalnymi, w szczególności Enterprise JavaBeans (zwłaszcza w implementacji wcześniejszej niż EJB3).

Co zwraca? chgw

7. Za co odpowiedzialne są adnotacje:

Adnotacje to metadane (a więc dane o danych — w naszym przypadku dane o kodzie), która dostarczają nam takich informacji *o programie*, które nie są częścią tego programu (same w sobie). Adnotacje nie mają bezpośredniego wpływu na kod, który nimi oznaczamy.

a. @SpringBootApplication - jest często umieszczana na głównej klasie i domyślnie definiuje podstawowy "pakiet wyszukiwania" dla niektórych elementów. Na przykład, jeśli piszesz aplikację JPA, pakiet klasy adnotowanej @SpringBootApplication jest używany do wyszukiwania elementów @Entity. Użycie pakietu głównego umożliwia również skanowanie komponentu do zastosowania tylko w twoim projekcie.

b. @RestController - adnotacja stereotypów, mówi Spring, aby wyrenderować ciąg bezpośrednio z powrotem do klienta (specyficzne dla Spring boot)

c. @RequestMapping - mapuje adres URL i/lub metodę HTTP na metodę kontrolera (specyficzne dla Spring boot)

d. @Autowired - może mieć zastosowanie do metod setter, metod nie-setterowych, konstruktora i właściwości. Deklaruje konstruktor, pole, metodę setter lub metodę konfiguracji, która ma być autowiredowana według typu. Elementy opatrzone adnotacją @Autowired nie muszą być publiczne.

8. Na czym polega Dependency Injection

„Wstrzyknięte zależności”

Każda aplikacja oparta na Javie ma kilka obiektów, które współpracują ze sobą, aby przedstawić to, co użytkownik końcowy widzi jako działającą aplikację. Podczas pisania złożonej aplikacji Java klasy aplikacji powinny być możliwie jak najbardziej niezależne od innych klas Java, aby zwiększyć możliwość ponownego wykorzystania tych klas i przetestować je niezależnie od innych klas podczas testowania jednostkowego. Dependency Injection (lub czasami zwane okablowaniem) pomaga w klejeniu tych klas razem, zachowując przy tym niezależność.

9. Tomcat, Jetty, undertow – czym są te aplikacje, do czego służą, w jaki sposób z nich korzystamy w Spring boot

Framework obsługuje trzy różne typy webowych serwerów: Tomcat (domyślnie), Jetty i Undertow. Ponieważ Tomcat jest domyślny, nie ma potrzeby wykonywania domyślnej implementacji, aby używać Tomcat. Możesz zmienić wersję używanego Tomcat lub zmienić właściwości w plikach pom.xml lub application.properties.

Aby zmienić osadzony kontener serwletu na Jetty, musisz edytować plik pom, aby usunąć zależność Tomcat i dodać Jetty. Podobnie dla Undertow.

Tomcat – kontener aplikacji webowych (ang. „web container”) rozwijany w ramach projektu Apache. Jako kontener aplikacji jest serwerem, który umożliwia uruchamianie aplikacji internetowych w technologiach Java Servlets i Java Server Pages (*JSP*). Jest to jeden z bardziej popularnych kontenerów Web. Tomcat jest wykorzystywany w takich serwerach aplikacji JEE (J2EE) jak Apache Geronimo. Jest również bardzo popularnym kontenerem dla samodzielnych aplikacji (niewymagających pełnego serwera aplikacji) pisanych w środowisku Spring Framework.

Jetty – otwarty serwer www oraz kontener serwletów Javy. W całości napisany w Javie i rozpowszechniany na licencji Apache 2.0.

Undertow – to elastyczny serwer WWW, napisany w Javie, zapewniający zarówno blokujące, jak i nie blokujące API oparte na NIO. Undertow ma architekturę opartą na komponentach, która pozwala na zbudowanie serwera WWW poprzez połączenie małych funkcji do obsługi pojedynczych obiektów.

10. Do czego służy Spring Boot Actuator

Actuator (siłownik) jest terminem produkcyjnym, który odnosi się do mechanicznego urządzenia do przenoszenia lub kontrolowania czegoś. Actuary mogą generować dużą ilość ruchu z niewielkiej zmiany. Punkty końcowe aktuatora umożliwiają monitorowanie i interakcję z aplikacją.

11. W jaki sposób łapiemy wyjątki w Spring Boot?