

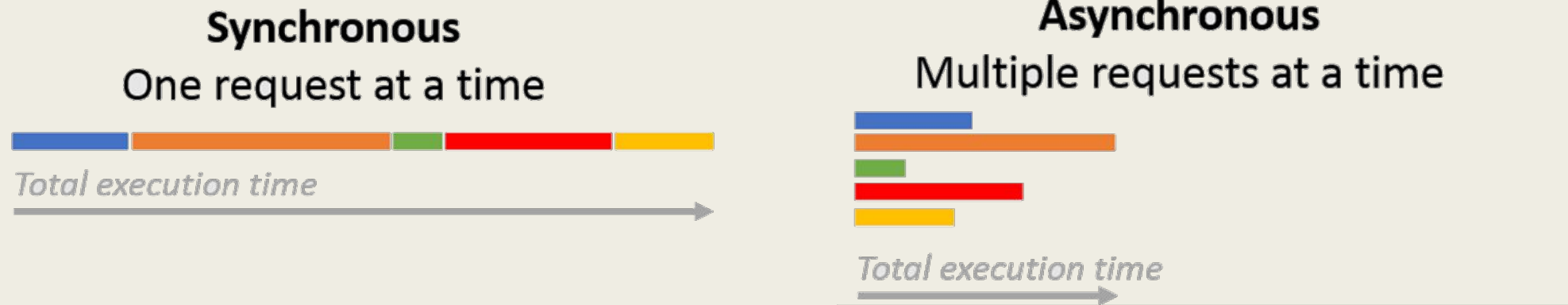


WZORCE
REAKTYWNE



Procesy

Synchroniczne vs. Asynchroniczne



Z rozwojem aplikacji internetowych i liczbą zapytań jakie muszą obsłużyć, szybka odpowiedź na żądanie HTTP, przez procesy synchroniczne stała się nieosiągalna. Problem ten rozwiązały procesy asynchroniczne.

PROGRAMOWANIE REAKTYWNE – ZASADA DZIAŁANIA

Tutaj interakcja między aplikacją kliencką a serwerem odbywa się za pomocą asynchronicznego strumienia danych. Możemy zaimplementować zasób, który będzie reagował z tym strumieniem danych, gdy przepływ danych jest odbierany. Każde zadanie działa wtedy jako funkcja i nie obsługuje zmiennych poza jego zakresem, dzięki czemu funkcja ta może być wykonywana kilka razy w sposób równoległy bez żadnych skutków ubocznych.

PROGRAMOWANIE REAKTYWNE - CECHY

- Elastyczne (ang. Elastic) – aplikacja używa wielu rdzeni i serwerów aby przetworzyć żądania,
- Odporna (ang. Resilient) – aplikacje reagują na błędy i są odporne na usterki w oprogramowaniu, sprzęcie lub sieci
- Oparte na wiadomościach (ang. Message-driven) – aplikacje opierają się na menedżerach zdarzeń (ang. event managers), które są asynchroniczne i nie blokują wątku
- Responsywne (ang. Responsive) – reagowanie na działania użytkowników. Aplikacja oferuje bogate interakcje przy bardzo niskim czasie oczekiwania.

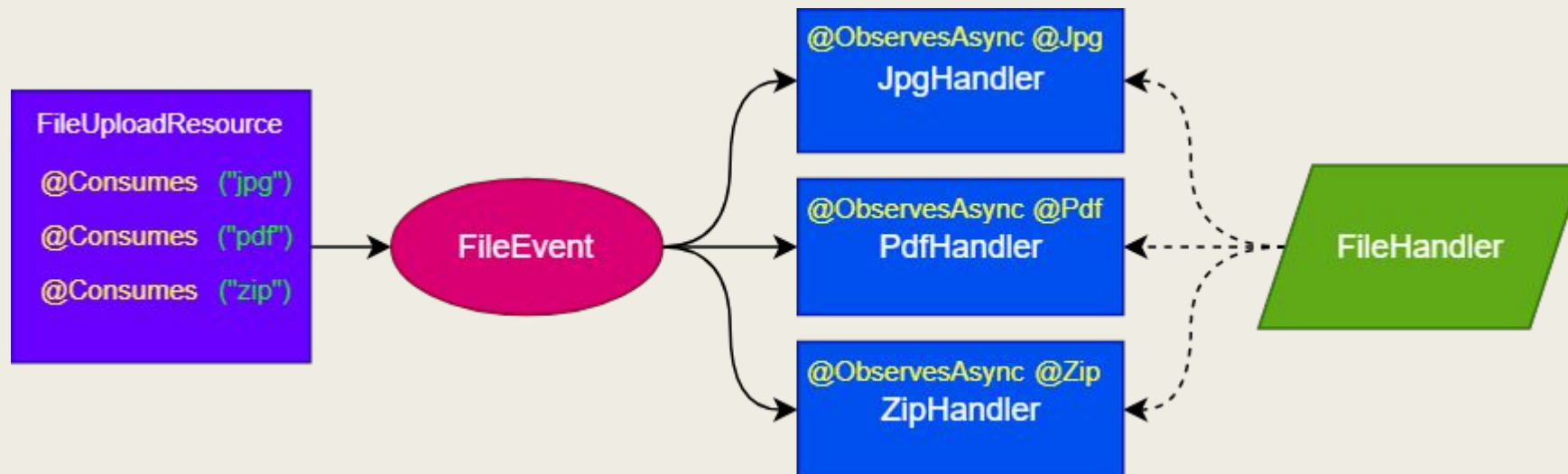
JAVA EE 8

Przykłady wzorców reaktywnych

- Zdarzenia w CDI (Contexts and Dependency Injection),
- Asynchroniczne metody EJB (Enterprise JavaBeans),
- Asynchroniczne serwisy REST (Representational State Transfer).

CDI EVENT

- Połączenie warstwy internetowej z warstwą transakcyjną platformy Java EE.
- Kod może uruchomić zdarzenie w sposób synchroniczny i blokujący lub asynchroniczny i nieblokujący
- Wzorzec obserwatora (ang. Observer pattern) - przekazanie Event do dowolnego komponentu działającego w sposób asynchroniczny



EJB - ENTERPRISE JAVA BEANS

W komponentach EJB można tworzyć metody asynchroniczne. Jest to mechanizm, który pozwala klientowi wywołanie metody i otrzymanie odpowiedzi od serwera jak tylko metoda zostanie wywołana.

Cechy:

- Mogą zwracać obiekt klasy `java.util.concurrent.Future`
- Klient może sterować wykonaniem metody (np. może je anulować)
- Klient może weryfikować postęp wykonywania metody.

ZDARZENIE CDI vs METODA EJB

CDI

- zasada jeden do wielu
- wykonywanie zadań z racji wykorzystania wzorca obserwatora, może się opóźnić
- umożliwia zastosowanie metody wywołania zwrotnego

EJB

- zasada jeden do jednego
- jest najczęściej uruchamiana od razu
- nie ma możliwości zastosowania metody wywołania zwrotnego

IMPLEMENTACJA KOMPONENTU EJB

```
@Stateless
public class PdfHandler {

    @Asynchronous
    public Future<String> handler(FileBean file) throws IOException {

        return new AsyncResult(
            FileSystemUtils.save(
                file.getFile(),
                "pdf",
                "pdf_" + new Date().getTime() + ".pdf"));
    }
}
```

IMPLEMENTACJA KLASY OBSŁUGUJĄCEJ USŁUGI REST

```
@Path("upload")
public class FileUploadResource {

    @Inject
    private PdfHandler pdfHandler;

    @Inject
    private JpgHandler jpgHandler;

    @Inject
    private ZipHandler zipHandler;

    @Consumes("application/pdf")
    @POST
    public Response uploadPdf(File file) throws IOException {

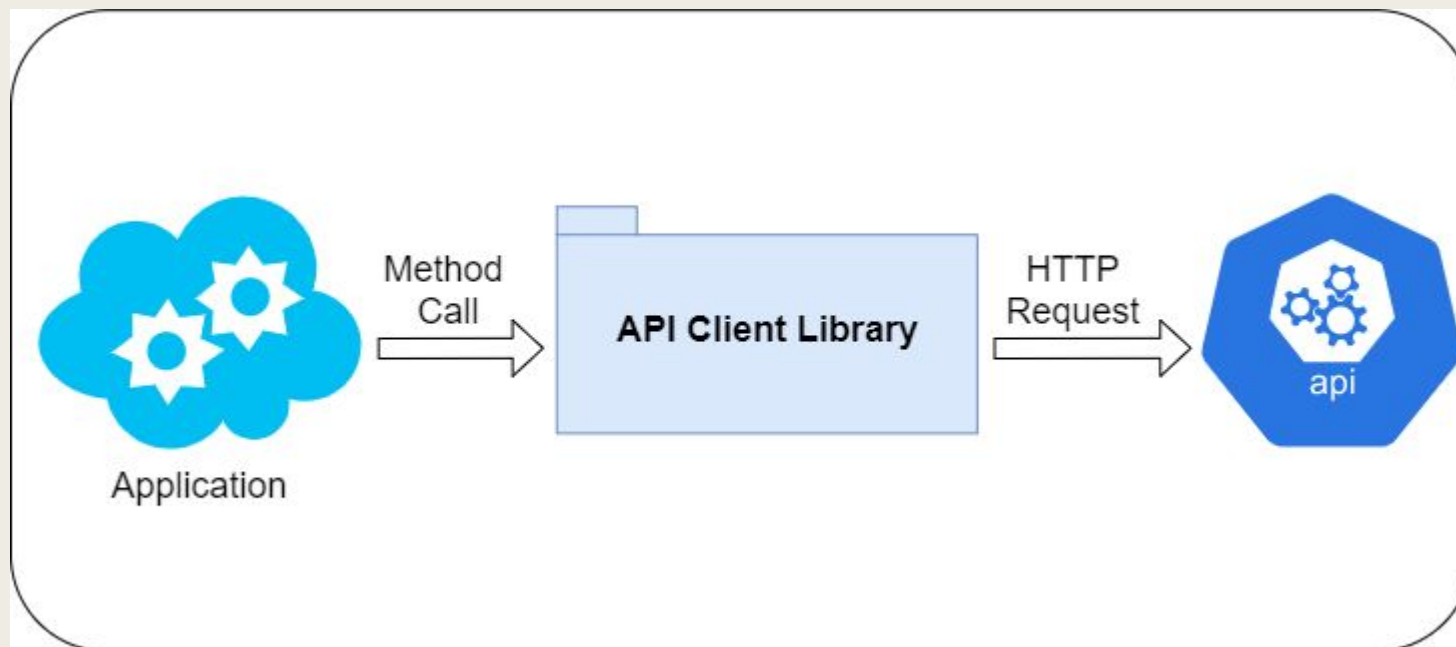
        FileBean fileBean = new FileBean(file, "pdf");

        pdfHandler.handler(fileBean);

        return Response.ok().build();
    }
}
```

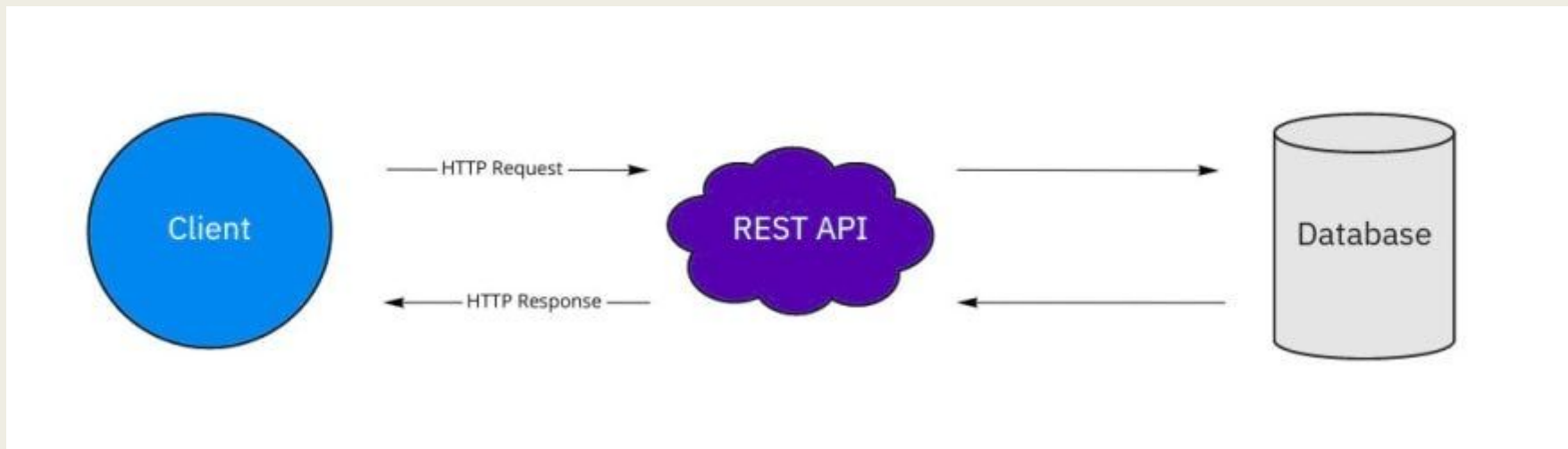
KLIENT API

API definiuje jak użytkownik może skomunikować się z systemem, reguły określające jak użytkownik może uzyskać dostęp do zasobów oraz w jakiej postaci je otrzymuje. Natomiast REST to styl architektury definiujący kształt API.



REST API

REST (REpresentational State Transfer) to styl architektury oprogramowania, którego charakterystycznymi cechami są: zorientowanie na zasoby, jednorodny interfejs oraz bezstanowość usług. Styl ten bardzo często jest stosowany w systemach rozproszonych, dzięki czemu przyczynił się do bardzo dynamicznego ich rozwoju w ostatnich latach. Został on zaprezentowany przez Roya Fieldinga roku 2000.



IMPLEMENTACJA ASYNCHRONICZNEGO REST

REST nie jest specyfikacją, więc istnieje pewna dowolność w implementowaniu RESTa. Istnieje jednak całkiem sporo różnego rodzaju reguł, które pomagają wprowadzić dobre praktyki w jego implementacji.

Do sklasyfikowania API restowego możemy posłużyć się modelem dojrzałości Richardsona, który dzieli REST API na 4 różne poziomy:

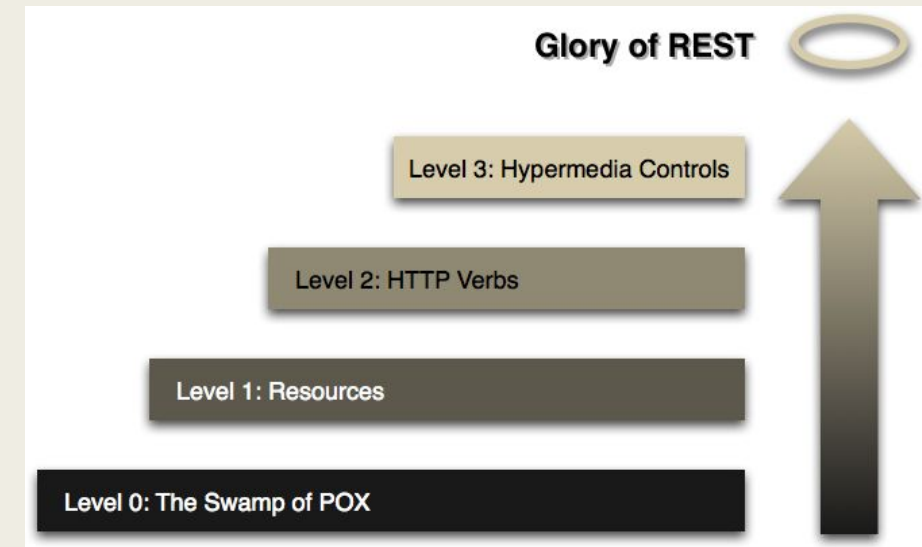
Poziom 0 - API implementujące ten poziom, nie korzysta z wielu adresów URI. Zwykle jest to np. 1 adres URI na cały serwis,

Poziom 1 - w tym przypadku, usługa zorientowana jest na zasób. Wykorzystywane są adresy URI do rozróżnienia zasobów. Zwykle używane są tylko metody GET i POST,

Poziom 2 - W poziomie tym używane są dodatkowe metody http: GET, POST, PUT, PATCH, DELETE. Metody te powinny być używane zgodnie ze swoim przeznaczeniem. Poza tym API powinno korzystać z odpowiednich statusów http np. 200 - udane połączenie, 429 - za dużo zapytań lub 400 - bad request,

Poziom 3 - zakłada obsługę HATEOAS (Hypermedia as the Engine of Application State). Dzięki czemu w odpowiedzi wysyłane są dodatkowe linki pozwalające poruszać się po API.

Mechanizm asynchronicznej usługi REST polega na tym, że sterowanie wraca do procesu tuż po wysłaniu żądania, więc klient nie musi czekać na jego zakończenie.



PRZYKŁAD

Wdrażanie asynchronicznych klientów REST może być zaawansowaną strategią dla aplikacji opartych na mikrousługach. Klienci asynchroniczni nie muszą czekać na odpowiedź, aby kontynuować pracę. W związku z tym mikrousługi w aplikacji mogą nadal przetwarzać i wysyłać dane, nawet jeśli jedna z ich usług partnerskich ma problemy. Ta funkcja zapewnia użytkownikowi bardziej niezawodną usługę i może być szczególnie cenna w przypadkach, gdy dostępność usługi jest niska lub przeciążona popytem.

Na przykład klienci asynchroniczni mogą być zaimplementowani w aplikacji usługi turystycznej, która wykonuje połączenia REST z usługą linii lotniczych, usługą hotelową i usługą wynajmu samochodów. Jeśli usługa hotelowa doświadcza opóźnienia, linia lotnicza i wypożyczalnia samochodów są w stanie kontynuować pracę.