Архитектурни стилови на веб апликацијата

# Архитектура за проток на податоци

Веќе сме користеле концептот на цевка и филтер во архитектурата за проток на податоци, специфично во контекстот на примање на податоци од OpenStreetMap. Преку овој модел, податоците се филтрираат и се издвојуваат оние кои се потребни, а потоа се зачувуваат во .csv фајлови. Главната цел на оваа архитектура е да обезбеди повторна искористливост на кодот.

Клиент-сервер архитектура  
Клиент-сервер архитектурата е модел на софтверска архитектура каде што функционалноста на системот е поделена помеѓу два основни дела: клиентската страна и серверската страна. Клиентите се одговорни за корисничкиот интерфејс и обработката на кориснички барања, додека серверите ги обработуваат барањата и управуваат со ресурсите и податоците. Комуникацијата помеѓу клиентите и серверите се извршува преку мрежа, каде клиентите бараат услуги или ресурси од серверите, а серверите ги обезбедуваат и одговараат на барањата. Оваа архитектура обезбедува модуларност, скалабилност и подобрена управување со системите. Протокол за комуникација е HTTP.

# Сервисно ориентирана архитектура (Service-oriented architecture)

Сервисно-ориентираната архитектура (СОА) е метод за развој на софтвер кој користи компоненти на софтвер наречени сервиси за создавање на бизнис апликации. Секој сервис обезбедува бизнис логика, а сервисите можат да комуницираат помеѓу себе низ различни платформи и јазици. Развивачите го користат СОА за повторна употреба на сервиси во различни системи или за комбинирање на повеќе независни сервиси за извршување на комплексни задачи.

# Resource-Oriented Architecture (REST-full)

Нашата апликација се заснова на SpringBoot како back-end и React како front-end дел. Spring boot го користиме како RESTtful web service, кој комуницира со React-тот преку Rest Controllers. Во самоиот Spring boot имаме:

1. **Адресабилност (Addressability):** Податоците се изложени на сервисните крајности, овозможувајќи интеракција преку URI.
2. **Безсостојбен (Statelessness):** При секој HTTP повик да ја содржи целосната потребна информација.
3. **Поврзаност (Connectedness):** Се вклучуваат и податоци и линкови до други ресурси. Restful веб сервисите следат HATEOAS (Hypermedia As The Engine Of Application State) принцип. Со ова се овозможува флексибилност и адаптивност на системот.

# Централизирана архитектура

Централизираната архитектура во нашиот контекст е најдобро илустрирана преку примерот на базата на податоци. Во нашата апликација, .csv фајловите ги импортираме во postgres база, произлезни од филтрирањето, и таа ја користиме како централна база на податоци. Преку неа добиваме информации за точната локација на културно-историските наследства. Главната цел на оваа архитектура е постигнување на поголем интегритет на податоците, како и обезбедување на надежност и сигурност. Воедно, клиентите стануваат независни од системот, што доприноси кон побаруваната автономија.