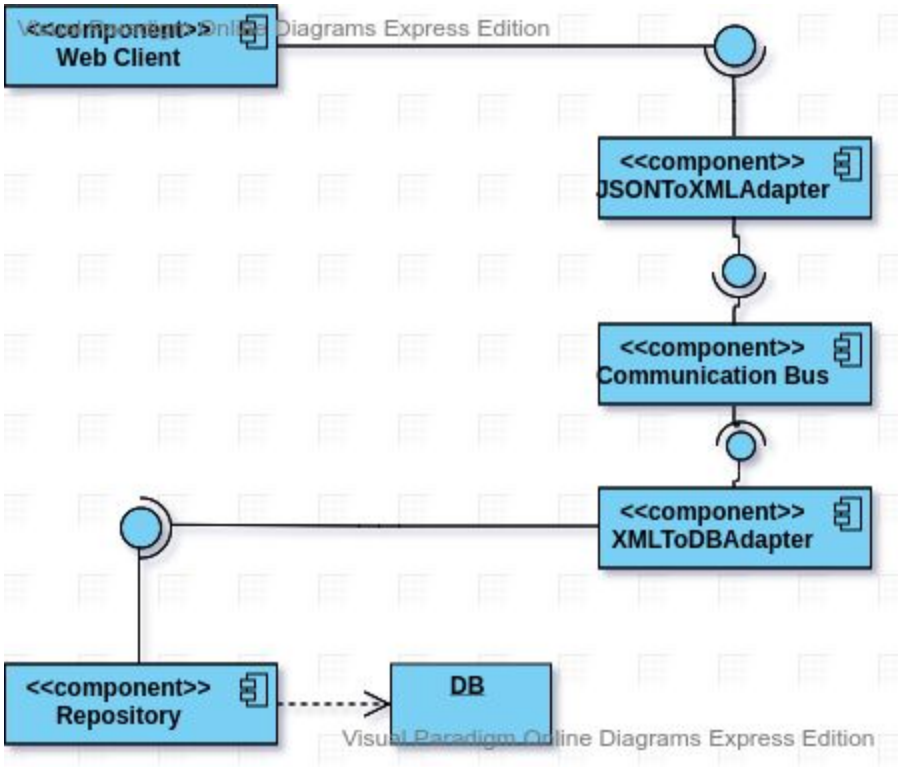
Развој електроенергетског софтвера
Факултет техничких наука Нови Сад
Школска година: 2019/2020
Асистент: Михаило Васиљевић
Пројекат додељен: 24.04.2020.

Communication Bus

Потребно је направити дизајн система, архитектуру система, имплементира и истестирати решење који симулира рад и комуникацију *Communication Bus* модула.



Начин рада система

Web client компонента добија од стране својих клијената неки од захтева:
GET, POST, PATCH, DELETE праћен одговорајућом именицом.
На пример: GET /resurs/1. Она тај захтев добија у одговарајућем JSON формату:

```
{
   "verb": "GET",
   "noun": "/resurs/1",
   "query": "name='pera'; type=1",
   "fields": "id; name; surname"
}
```

Сама компонента има задатак да провери да ли је захтев добро форматиран (да ли прослеђени глагол постоји, да ли су вредности поља qyeру и фиелдс форматирана на одговарајући начин). Поља query и fields нису обавезна, она додатно филтрирају ресурсе.

Како ***Communcation Bus*** не разуме JSON као формат података, већ искључиво XML, неопхопдно је захтев претвори у XML и то је сврха JSONToXMLAdapter компоненте. XML од претходног JSON захтева би био:

```
<request>
  <verb>GET</verb>
  <noun>/resurs/1</noun>
  <query>name='pera';type=1</query>
  <fields>id; name; surname</fields>
</request>
```

Communication bus добијени захтев само прослеђује својој адаптер компоненти XMLToDBAdapter која треба овај XML да претвори у одговарајући SQL упит (изглед упита ће да зависи од базе коју одаберете):

```
SELECT
 id,
 name,
 surname
FROM resurs
WHERE id = 1
AND name = 'pera'
AND surnatypeme = 1
```

Repository компонента прима захтев у овом облику и извршава га над базом података, добија одговарајуци одговор и добијени одогвор прослеђује XMLToDBAdapter компоненти да га претвори у одговарајући XML, сваки одговор мора да поседује поља: STATUS, STATUS_CODE и PAYLOAD. Дозвољени статуси су: REJECTED, BAD_FORMAT, SUCCESS, а статус кодови: 3000, 5000 и 2000 респективно. Payload би требало да садржи добијени ресурс односно у случају примеру XML поља id, name и surname у случају SUCCESS кода односно error-message поље са поруком о грешци. Тако добијени XML Communication bus прослеђује својој JSONToXMLAdapter компоненти кој га трансформише у JSON. Затим се одговор прослеђује Web Client компоненти која га враћа својим клијентима. У случају да захтев није био добро форматиран Web Client одмах враћа BAD_FORMAT 5000 са одговарајућом поруком о грешци, не прослеђује захтев даље.

Напомена:

Захтев поред поља наведених у примеру може да поседује и следећа поља: connectedTo са списком ид-ова повезаних ресурса(ид=1; ид=3), при чему је ово додатни филтер, на пример ако постоји више ресурса са именом Петар врати ми оног који је у вези са ресурсима Клавир и Одбојка, као и connectedType где се наводе ид-ови одговарајућих типова, при чему је и ово додатни филтер и користан је кад на пример имам више ресурса са именом Петар а потребни

су ми само они који постоје у типу везе родитељ. Када клијент Web Client компоненте креира захтев обавезан је да унесе само verb и noun, све остало представља само додатни филтер који може, а не мора да унесе.

Модел података

Ресурс:

- Id
- Naziv
- Ime
- Opis (у JSON формату, уколико изабрана база подржава JSON као тип података колоне)
- Tip

Тип:

- Id
- Naziv

Веза:

- Id
- Id provog resursa
- Id drugog resursa
- Tip

Тип везе:

- Id
- Naziv

Критеријум оцењивања

1. Дизајн и архитектура решења
2. Коришћење *Scrum* методологије развоја – дефинисање *user story*-а и таскова, планирање и естимација
3. Имплементација решења
4. CI циклус
    a. Build
    b. Unit Tests
    c. Покривеност кода тестовима