

Дизајн и архитектура на софтвер

Домашна работа 4

Во четвртата домашна работа, треба да го рефакторирате кодот од третата домашна работа, да ги префрлите дел од функционалностите на апликацијата во АПИ/микросервиси и да ја подготвите апликацијата и микросервисите за работа во контејнер. На крај, треба да ја инсталирате апликацијата и сервисите во облак.

1 Рефакторирање на кодот

Рефакторирајте го кодот на вашата апликација да следи **најмалку еден од софтверските шаблони** дискутирани на предавањата. Посавете документ со објаснување кој шаблон сте го имплементирале и зошто. Рефакторираниот код треба да:

- **Биде јасен:** Кодот треба да биде лесен за читање и разбирање.
- **Биде доследен:** Користете ист стил на именување за променливи и методи низ целиот код.
- **Има добри имиња:** Променливите и методите треба да имаат јасни и значајни имиња.
- **Биде добро документиран:** Имињата треба да го објаснуваат кодот. Ако не е така, додадете кратки коментари за да разјасните што прави кодот или да објасните нејасни имиња на методи или параметри.
- **Биде лесен за одржување:** Кодот треба да биде едноставен за ажурирање или подобрување во иднина.
- **Биде повторно употреблив:** Напишете код што може да се користи на други места во апликацијата.
- **Избегнува повторување:** Отстранете дупликатен код правејќи ја апликацијата го пофлексибилна и модуларна.
- **Биде ефикасен:** Користете алгоритми кои се едноставни и брзи, избегнувајќи непотребна сложеност.

2 АПИ/Микросервиси

Преработете некои функции од вашата апликација како микросервиси кои комуницираат преку АПИ. Микросервисите се мали, независни делови од апликацијата кои можат да работат самостојно. Земете неколку функции од вашата апликација, изградете ги како посебни програми и осигурајте се дека можат да функционираат самостојно. **Секој микросервис треба да биде создаден како свој проект.** Потоа, овозможете комуникација помеѓу микросервисите и главната апликација користејќи АПИ.

Можете да ги следите овие туторијали за примери како да го направите ова користејќи Spring. Ако користите друга технологија, можете да ги прилагодите чекорите:

- <https://spring.io/blog/2015/07/14/microservices-with-spring>
- <https://www.javatpoint.com/microservices>
- <https://developer.okta.com/blog/2019/05/22/java-microservices-spring-boot-spring-cloud>



3 Контејнеризација и инсталација во облак

Подгответе ја вашата апликација за деплојирање преку контејнеризација и деплојирање во облак. Контејнерите се самостојни единици кои ја вклучуваат вашата апликација и сите нејзини зависности, обезбедувајќи конзистентност во различни опкружувања. Користете алатка како што е **Docker** за пакување на вашата апликација во контејнер.

За повеќе информации за користење Docker, погледнете:

- <https://docs.docker.com/get-started/>

Дополнително, следниот водич обезбедува пример како да се контејнеризира микросервис со Docker:

- <https://azure.microsoft.com/en-us/free/students/>
- <https://aws.amazon.com/education/awseducate/>

Имајте на ум дека ќе мора да почекате неколку денови за потврдата на вашиот статус на студент, која е потребна за да ги користите сервисите.

Кодот за четвртата домашна поставете го на GitHub во папка именувана Домашна 4.

Важно: Осигурете се дека верзијата на апликацијата за третата домашна работа останува непроменета. За да го направите ова, избегнувајте да работите на гранката поврзана со платформата каде што е деплојирана апликацијата, или исклучете ги автоматските деплојменти.

Објавете ја обновената апликација за четвртата домашна работа на нов линк и вклучете го овој линк во *About* секцијата на вашиот GitHub репозиториум.

Рокот за изработка на третата домашна работа е до **19 јануари 2025**. Имајте во предвид дека овој рок е конечен и нема да биде продолжен.