

Listing glavnih delova implementacije čuvanja primarne instance

- Kreiranjem primarnog entiteta (Bill) preko Web Klijenta, unutar kontrolera za primarni entitet (BillController), u akciji CreateNew konektujem se na WCF servis koji sam implementirao u Entity Handler worker roli i prosledjujem mu objekat primarnog entiteta.

```
// CREATE
[HttpPost]
public ActionResult CreateNew(string rowKey, DateTime creationTime, int numOfProducts)
{
    BillEntity be = new BillEntity(rowKey);
    be.CreationTime = creationTime;
    be.NumOfProducts = numOfProducts;

    try
    {
        foreach (var instance in RoleEnvironment.Roles["EntityHandler_WorkerRole"].Instances)
        {
            if (instance.Id.Contains("_IN_0") || instance.Id.Contains("_IN_2"))
            {
                NetTcpBinding binding = new NetTcpBinding();
                string endpoint = instance.InstanceEndpoints["InternalWeb"].IPEndpoint.ToString();
                ChannelFactory<IEntityOperationRequest> factory = new ChannelFactory<IEntityOperationRequest>(binding, new EndpointAddress(String.Format("net.tcp://localhost:{0}/InternalWeb", endpoint)));
                IEntityOperationRequest proxy = factory.CreateChannel();

                proxy.CreateBill(be);

                return RedirectToAction("Index");
            }
        }
    }

    return RedirectToAction("Index");
}
```

- Unutar same implementacije WCF servisa kod EntityHandler worker role, pozivam metode vezane za manipulaciju Table Storage-om. Jedna od njih je i za cuvanje primarnog entiteta u Table Storage (AddBill).

```
namespace EntityHandler_WorkerRole
{
    public class EntityOperationRequestProvider : IEntityOperationRequest
    {
        BillDataRepository repoBill = new BillDataRepository();
        EntryDataRepository repoEntry = new EntryDataRepository();
        ProductDataRepository repoProduct = new ProductDataRepository();

        // IMPLEMENTACIJE ZA BILL

        public void CreateBill(BillEntity bill)
        {
            repoBill.AddBill(bill);
        }
    }
}
```

Navesti nazive resursa u Cloud-u (nazive tabela, kontejnera i sl.)

- Naziv tabele za Bill entitet - **BillTable**
- Naziv tabele za Entry entitet - **EntryTable**
- Naziv tabele za Product entitet - **ProductTable**

- Naziv Queue-a - **stanje**

Opisati nekoliko test slučajeva koji se mogu izvršiti kako bi se svi aspekti sistema pokrili i testirali korak po korak npr jedan test za primarne instance:

Kakav zahtev se šalje : Web klijent salje zahtev za izvršenje neke od CRUD operacija nad posmatranim entitetom, putem internal WCF komunikacije sa EntityHandler worker rolom.

Koja instanca i zašto preuzima zahtev: U slučaju primarnog entiteta zahtev preuzima ili prva ili treća instanca EntityHandler worker role, to se postize preko kalse RoleEnvironment, proverom da li instanca u svom id-u ima "_IN_0" za slučaj prve instance i "_IN_2" za slučaj treće instance.

Instanca koja je preuzela zahtev poziva implementiranu metodu iz Commona, za datu prosledjenu CRUD operaciju, a unutar implementacije te metode se poziva neka od metoda za manipulaciju nad Table Storage-om

U isto vreme, HealthService worker rola putem internal WCF- komunikacije propituje stanje prve instance EntityHandler worker role i dobijen odgovor stavlja u Queue.

Poruke iz Queue se dalje preuzimaju u Notifier worker roli, koja dalje salje tu poruku na HealthMonitoring Console App gde se ta poruka prikazuje.

David Prelić PR129/2017